Tampereen yliopisto

Ilona Vänni

POIKKEAVIEN HAVAINTOJEN TUNNISTUS KONEOPPIMISMENETELMIN

Informaatioteknologian ja viestinnän tiedekunta
Pro gradu -tutkielma
12 2019

TIIVISTELMÄ

Ilona Vänni: Poikkeavien havaintojen tunnistus koneoppimismenetelmin
Pro gradu -tutkielma
Tampereen yliopisto
Computational Big Data Analytics
12 2019

Poikkeavien havaintojen tunnistus on tärkeä osa datalähtöisiä prosesseja. Poikkeavien havaintojen tunnistuksen avulla aineistojen laadukkuus saadaan taattua ja toisaalta mielenkiinnon kohteena olevat poikkeavuudet aineiston normaalista rakenteesta tunnistettua. Poikkeavuuksien tunnistamiseen käytetään erilaisia perinteisiä tilastollisia testejä, mutta näiden rinnalle on noussut myös erilaisia koneoppimiseen pohjautuvia menetelmiä. Koneoppimismenetelmien avulla pystytään tunnistamaan äärihavaintojen lisäksi erityyppisiä poikkeavuuksia. Myös poikkeavien havaintojen ryhmittymiä on mahdollista tunnistaa suuristakin aineistoista, jotka koostuvat sekatyyppisistä muuttujista.

Tässä tutkielmassa kartoitetaan koneoppimismenetelmien käyttöä poikkeavien havaintojen tunnistukseen tilastollisesta aineistosta. Koneoppimisen hyödyntämistä poikkeavuuksien tunnistuksessa on tutkittu eri sovellusalueilla ja todettu aineiston rakenteesta riippuen sekä ohjatun että ohjaamattoman oppimisen menetelmien sopivan tehtävään. Koneoppimismenetelmiä on kehitetty myös erityisesti tätä tehtävää varten. Vaikka poikkeavuuksien tunnistuksesta koneoppimismenetelmin on laajasti tutkimusnäyttöä, ei niiden soveltamista tilastollisen aineiston tapauksessa ole juurikaan kartoitettu.

Tutkimusaineisto koostetaan Suomen Pankin tuottamasta luottolaitosten tase- ja korkotilastosta. Tutkielman tavoitteena on kartoittaa koneoppimismenetelmien hyödyntämistä osana rahoitustilastojen laadunvalvontaprosessia. Tutkielmassa harkitaan kahdeksaa eri ohjatun ja ohjaamattoman oppimisen menetelmää ja mallien hyvyyttä tarkastellaan ensisijaisesti tilastolaadinnan näkökulmasta. Osittavat ohjaamattoman oppimisen mallit osoittautuivat hyödyllisiksi aineiston rakenteen hahmottamisessa, mutta poikkeavien havaintojen tunnistamisessa näiden menetelmien ennustekyky oli heikko. Ohjatun oppimisen menetelmistä päätöspuupohjaiset algoritmit onnistuivat normaaleiden havaintojen ennustamisessa hyvin, mutta myös suuri osa poikkeavista havainnoista luokiteltiin normaaleiksi havainnoiksi. Päätöspuupohjaisista menetelmistä etenkin eristysmetsä on kuitenkin harkitsemisen arvoinen menetelmä osaksi tilaston laadunvalvontaprosessia, sillä poikkeavuusarvon alarajaa nostamalla voidaan poimia vain hyvin suurella todennäköisyydellä poikkeavuuksia olevat havainnot, vaikkakin tällöin jää myös aitoja poikkeuksia tunnistamatta. Ylitse muiden menetelmien poikkeavien havaintojen tunnistuksessa onnistui k:n lähimmän naapurin menetelmä, jonka ennustetarkkuus sekä aitojen poikkeavuuksien että normaalien havaintojen suhteen oli erityisen korkea.

Tutkielman tulosten pohjalta koneoppimismenetelmiä olisi hyödyllistä harkita osaksi tilastojen laadintaprosessia. Luottolaitosten tase- ja korkotilaston poikkeavien havaintojen tunnistukseen k:n lähimmän naapurin menetelmä soveltuu erinomaisesti, ja menetelmän käyttö osana laadunvalvontaa potentiaalisesti tehostaisi prosessia sekä osaltaan edistäisi tilastoaineiston laadukkuutta. Toisaalta ohjaamattoman oppimisen menetelmiä voisi olla hedelmällistä käyttää apuna tilastoaineiston analysoinnissa niiden paljastaessa aineistosta rakenteita ja säännönmukaisuuksia, joita on vaikea tunnistaa ilman koneoppimismallintamista.

Avainsanat: Koneoppiminen, rahoitustilasto, poikkeavuuksien tunnistus, k:n lähimmän naapurin menetelmä, päätöspuu, satunnaismetsä, eristysmetsä, DBSCAN, K-means

ABSTRACT

Ilona Vänni: Machine learning based outlier detection
Master thesis
Tampere University
Computational Big Data Analytics
12 2019

Detecting outliers is an important part of data-driven processes. By identifying outliers, the quality of the data can be guaranteed, and on the other hand, outliers can be identified when the main interest are the anomalies of the structure. Traditionally different statistical tests are being used to identify outliers, but along with these, different types of machine learning methods have emerged. In addition to extreme values, machine learning methods can detect different types of outliers. With machine learning techniques it is also possible to identify clustered outliers from large data sets consisting of mixed-type variables.

This thesis explores the use of machine learning methods to identify outliers in statistical data. Machine learning in the detection of outliers has been studied in different application areas and both, supervised and unsupervised learning methods, have found to be useful, depending on the structure of the data. Machine learning methods have also been developed specifically for this task. Although there is widespread research evidence about the outlier detection with machine learning methods, there is little evidence of their power in case of statistical data.

The research data is gathered from the balance sheet and interest rate statistics of monetary financial institutions produced by the Bank of Finland. The aim of this thesis is to map the utilization of machine learning methods as part of the quality control process of financial statistics. Eight different methods of supervised and unsupervised learning are considered in the thesis, and the goodness of models is considered primarily from the perspective of statistics production. Partitional unsupervised methods proved to be useful in understanding the structure of the statistical data, although the predictive power of these methods was poor in identifying outliers. From supervised methods, decision tree-based algorithms performed well in predicting the correct class, although a large proportion of outliers were also classified as normal observations. Of the decision tree-based methods, isolation forest, in particular, is a worthy method to consider as a part of the statistical quality control process. With isolation forest it is possible to capture observations with very high probability of being outlier, although in this case also some true outliers with lower probability remains unidentified. Superior to other considered methods, k nearest neighbors had extremely high predictive accuracy for both true outliers and normal observations.

Based on the findings of the thesis, it is useful to consider machine learning methods as a part of statistics production process. For the outlier detection in monetary financial institutions balance sheet and interest rate statistics, the k-nearest neighbor method is well suited, and its use as part of data quality control would potentially enhance the process and contribute to the quality of statistical data.

Keywords: Machine learning, financial statistics, outlier detection, k-nearest neighbors, decision tree, random forest, isolation forest, DBSCAN, K-means

ALKUSANAT

Tämä pro gradu –tutkielma toteutettiin yhteistyössä Suomen Pankin rahoitustilastotoimiston kanssa. Tutkimuskysymys ja –aineiston valinta suunniteltiin yhdessä Suomen Pankin tilastoyksikön asiantuntijoiden kanssa, ketkä ystävällisesti tarjosivat asiantuntevia kommentteja myös projektin aikana. Projekti oli äärimmäisen mielenkiintoinen ja opettavainen, ja auttoi osaltaan edistämään uusien menetelmien käyttöönottoa osana tilastolaadintaa.

Erityiskiitoksen haluan osoittaa ohjaajalleni Martti Juholalle asiantuntevasta ja motivoivasta ohjauksesta. Suuri kiitos kuuluu myös tutkielman toiselle tarkastajalle Kati Iltaselle. Niin ikään haluan kiittää Suomen Pankin rahoitustilastotoimistoa työn motivoinnista ja erityisesti tilastopäällikkö Elisabeth Flittneria rahoitustilastoaineiston käyttöoikeudesta tutkimustarkoitukseen. Lämmin kiitos myös läheisilleni tuesta ja kannustuksesta projektin aikana.

Helsingissä, 3.12.2019

Ilona Vänni

SISÄLLYSLUETTELO

TAULUKKOLUETTELO

KUVALUETTELO

Tampereen yliopisto

1. JOHDANTO

Poikkeavien havaintojen tunnistus on perinteinen tutkimusongelma, joka on viime vuosina saanut enenevissä määrin huomiota. Poikkeavuuksien tunnistuksesta on muodostunut yhä mielenkiintoisempi ja ajankohtaisempi kysymys digitalisoituneessa yhteiskunnassa, jossa päätöksenteko pohjautuu vahvasti dataan ja data-analyysiin. Riippuen sovellusalueesta motivaatio poikkeavuuksien tunnistamiseen vaihtelee. Luottokorttiyhtiö haluaa tunnistaa poikkeavan käytöksen luottokortin maksudatasta, jotta esimerkiksi varkaus havaittaisiin mahdollisimman varhaisessa vaiheessa. Analyytikot ja tilastojen laatijat taas haluavat karsia poikkeavat ääriarvot tilastoaineistoista ääriarvojen vaikuttaessa aineiston tunnuslukuihin normaaleja havaintoja voimakkaammin aiheuttaen mahdollista harhaa lukujen tulkintaan. Tutkijat osaltaan eivät halua poikkeavien havaintojen vaikuttavan mallinsa parametrien estimointiin. Poikkeavuuksien tunnistamista voidaan niin ikään käyttää eri tieteenaloilla mielenkiinnon kohteena olevien ilmiöiden havaitsemiseen, kuten lääketieteessä syöpäsolujen tunnistamiseen tai taloustieteessä kulutuskäyttäytymisen muutoksiin.

Myös keskuspankkitoiminnan näkökulmasta poikkeavuuksien tunnistaminen on tärkeä kysymys. Keskuspankkien rahapoliittinen päätöksenteko pohjautuu vahvasti dataan, ja niin ikään rahoitusvakauden seuranta on hyvin datalähtöistä. Yhdysvaltain keskuspankin pääjohtaja Jerome Powell on puhunut paljon datalähtöisestä päätöksenteosta ja datan laadukkuuden merkityksestä keskuspankin näkökulmasta. Kuten Powell totesi lokakuussa 2019 pitämässään puheessa: ”Hyvät päätökset vaativat hyvää data, mutta käsillä oleva data on harvoin niin hyvää kuin haluaisimme”. Datan laadukkuuden takaaminen on haasteellista, ja keskuspankit käyttävät paljon resursseja tuottaakseen osaltaan mahdollisimman laadukasta dataa. Yksi Euroopan keskuspankkijärjestelmään kuuluvien kansallisten keskuspankkien tehtävistä on kansallisten rahoitustilastojen tuottaminen rahapoliittisen päätöksenteon tueksi. Jotta tilastojen hyöty päätöksenteon tukena saadaan taattua, on tilastodatan laadukkuus erityisen tärkeää, koska virheellinen data johtaa potentiaalisesti vääriin johtopäätöksiin. Suomessa rahoitustilastojen laadinnasta vastaa Suomen Pankki. Suomen Pankin tilastoyksikkö tuottaa merkittävän määrän tilastoaineistoa kuukausittain, ja datan laadun varmistaminen on yksikön ydintehtäviä. Tilastoaineiston laadunvalvonnan parantaminen ja tehostaminen on yksikön jatkuva tavoite. Koneoppimisen hyödyntäminen osana prosessia mahdollisesti auttaisi merkittävästikin paitsi prosessin tehostamisessa, myös analyysin laadun parantamisessa.

Tässä tutkielmassa kartoitetaan erilaisten koneoppimismenetelmien suoriutumista Suomen Pankin rahoitustilastoaineiston poikkeavien havaintojen tunnistuksessa. Vaikka koneoppimismenetelmien hyödyntämistä poikkeavuuksien tunnistukseen on tutkittu laajasti, ei niiden soveltamista tilastollisen datan mallinuksessa ole juurikaan kirjallisuudessa käsitelty.

Tutkielma koostuu kuudesta pääluvusta. Johdannon jälkeen toisessa luvussa kartoitetaan lyhyesti poikkeavien havaintojen tunnistusta koneoppimismenetelmin käsittelevää kirjallisuutta. Kolmannessa luvussa käsitellään aiheeseen liittyvää teoriaa. Luvun alussa määritellään poikkeavan havainnon käsite ja poikkeavuuksien eri tyypit, jonka jälkeen käydään läpi koneoppimisen eri alaluokat. Tämän jälkeen käsitellään tarkemmalla tasolla ohjaamattoman ja ohjatun oppimisen algoritmien teoriaa. Tarkastelun alle on valittu menetelmiä, joiden on todettu soveltuvan poikkeavuuksien tunnistukseen. Luvussa neljä esitellään tutkimusaineisto sekä kuvataan aineistolle suoritetut esikäsittelyt. Viidennessä luvussa sovitetaan luvussa 3 esitetyt algoritmit tutkimusaineistoon ja analysoidaan mallien suorituskykyä sekä tuloksia. Viimeisessä luvussa tehdään yhteenveto tutkielman keskeisimmistä tuloksista ja analysoidaan tutkielman merkitystä tilastolaadinnan näkökulmasta. Luvun lopuksi ehdotetaan aiheita tuleviin tilastotuotannon koneoppimissovelluksiin.

2. KIRJALLISUUSKATSAUS

Poikkeavien havaintojen tunnistus on perinteisesti ollut tilastotieteen ja tilastollisen mallinnuksen keskeinen aineiston esikäsittelyn vaihe, koska poikkeavat arvot voivat vaikuttaa merkittävästikin etenkin pienten aineistojen tunnuslukuihin ja tilastollisten mallien estimointiin. Kuitenkin datan määrän ja käyttökohteiden lisääntyessä poikkeavien havaintojen tunnistuksesta on tullut laajemminkin kuin tilastotieteilijöiden keskuudessa kiinnostava tutkimuskysymys. Datalähtöisen analyysin ja päätöksenteon lisääntyessä digitalisaation myötä on ollut ja on yhä tärkeämpää, että aineisto päätöksenteon takana on mahdollisimman korkealaatuista. Joillakin sovellusalueilla poikkeavuudet ovat myös itsessään mielenkiinnonkohde ennemmin kuin normaaliksi luokiteltava data.

Tilastotieteessä poikkeavuuksien tunnistukseen on käytetty erilaisia tilastollisia testejä, kuten z-testiä tai John Tukeyn kehittämää aineiston kvartiileihin perustuvaa testiä, joka tunnetaan myös laatikkojanakuviona. Muuttujien välisten, multivariaattien, poikkeavuuksien tunnistukseen tilastotieteessä on yleisesti käytetty esimerkiksi Mahalanobiksen etäisyyteen perustuvaa testisuuretta. Monimuuttujaisten ja merkittävän suurien aineistojen yleistymisen myötä perinteisten testien rinnalle on kuitenkin noussut useita vaihtoehtoisia, koneoppimiseen perustuvia tekniikoita poikkeavuuksien tunnistamiseen.

Phua, Lee, Smith-Miles sekä Gayler (2010) toteavat kartoituksessaan luottokorttimaksudatan olevan yksi tutkituimmista poikkeavuuksien tunnistamisen sovellusalueista. Luottokorttiyhtiöillä on vahva intressi tunnistaa normaalista maksukäyttäytymisestä poikkeavat maksutapahtumat, jotta esimerkiksi väärinkäytökset voidaan tunnistaa mahdollisimman nopeasti. Patil, Nemade ja Soni (2018) toteavat satunnaismetsän olevan luokittelutarkkuudeltaan päätöspuupohjaista ID3-algoritmia sekä logistista regressiota parempi menetelmä luottokorttidatan poikkeavuuksien tunnistamiseen. Samaan tulokseen päätyvät Kurien ja Chikkamannur (2019) verratessaan satunnaismetsää ja logistista regressiota luottokorttimaksujen poikkeavuuksien tunnistamiseen. Myös muilla sovellusalueilla koneoppimista on sovellettu poikkeavuuksien havaitsemiseen, etenkin langattomien sensoriverkostojen poikkeavuuksia on mallinnettu mm. lähimpiin naapureihin pohjautuvilla menetelmillä sekä erilaisilla luokittelumenetelmillä, kuten tukivektorikoneilla ja Bayes-verkoilla. (Zhang, Meratnia ja Havinga, 2010) Niin ikään lääketieteessä poikkeavuuksien mallinnukseen on sovellettu koneoppimismenetelmiä. Crispi, Sahli, Monteagudo, Pacheco ja Falcon (2009) kartoittavat lääketieteellisten kuvien tunnistukseen käytettyjä menetelmiä, ja nostavat esiin etenkin tilastolliset mallit, neuroverkot, relevanssivektorikoneet sekä hybridimallit, jotka ovat edellä mainittujen yhdistelmämenetelmiä. Huang, Lu ja Duan (2012) toteavat kNN-menetelmän suoriutuvan potilaan hoitopolun poikkeavuuksien tunnistamisessa perinteisiä menetelmiä paremmin. Ijaz, Alfian, Syafrudin ja Rhee (2018) yhdistävät DBSCAN-pohjaisen poikkeavuuksien tunnistamisen satunnaismetsä- ja SMOTE-menetelmiin muodostaen hybridimallin tyypin 2 diabeteksen sekä kohonneen

verenpaineen sairastumisriskiin. Ijaz et al. toteavat DBSCAN-menetelmällä poikkeavuuksista puhdistetun aineiston parantavan ennustetarkkuutta mallinnettaessa sairastumisriskiä satunnaismetsämenetelmällä.

3. TEORIA

Poikkeavien havaintojen tunnistaminen on perinteinen tutkimusongelma ja erilaisten koneoppimismenetelmien soveltuvuutta tehtävään on kartoitettu laajasti. Käytettävä menetelmä valikoituu pääsääntöisesti käsiteltävän aineiston ominaisuuksien sekä käytettävissä olevan opetusaineiston perusteella. Poikkeavien havaintojen tunnistamista voidaan lähestyä yksi- tai kaksiluokkaisena luokitteluongelmana riippuen siitä, onko valmiiksi luokiteltua opetusaineistoa käytettävissä.

Eri menetelmillä on todettu olevan sekä etuja että varjopuolia tunnistettaessa poikkeavuuksia suurista aineistosta. Tässä luvussa esitellään poikkeavuuksien tunnistamiseen käytettyjä menetelmiä.

3.1 Poikkeava havainto

Erään käytetyimmistä poikkeavan havainnon määritelmistä esitti Douglas M. Hawkins vuonna 1980 teoksessaan Identification of Outliers. Hawkinsin määritelmän mukaan poikkeava havainto poikkeaa muusta aineistosta niin paljon, että sen kuvittelisi olevan eri mekanismin tuote kuin muut aineiston havainnot (Hawkins, 1980).

3.1.1 Poikkeavien havaintojen luokittelu

Poikkeavat havainnot voidaan jakaa kolmeen eri tyyppiluokkaan ominaisuuksiensa perusteella.

1. Tyypin I poikkeava havainto: Pistemäinen

Poikkeava havainto luokitellaan pistemäiseksi silloin, kun sen arvo poikkeaa merkittävästi muusta datajoukosta jonkun muuttujan suhteen. (Chandola, Banerjee & Kumar, 2009) Tällaiset ääriarvot ovat yleisin poikkeavuuksien tyyppi ja hyvin yleisiä myös tilastolaadinnassa. Tilastolaadinnassa pistemäiset poikkeavuudet osoittautuvat usein virheellisiksi arvoiksi, jotka aiheutuvat esimerkiksi pilkkuvirheistä raportin laadintavaiheessa. Pistemäiset poikkeavuudet tunnistetaan vertaamalla yksittäisen instanssin arvoa suhteessa muuhun, normaaliksi ajateltuun datajoukkoon.

2. Tyypin II poikkeava havainto: Kontekstuaalinen

Havainto voi olla poikkeava, vaikkei se saisi muusta datajoukosta merkittävästi poikkeavia arvoja minkään yksittäisen muuttujan suhteen. Kontekstuaaliseksi poikkeavuudeksi luokitellaan sellaiset havainnot, jotka eri muuttujien arvojen yhdistelmän suhteen poikkeavat muusta aineistosta. (Chandola et al., 2009)

3. Tyypin III poikkeava havainto: Kollektiivinen

Kollektiiviseksi poikkeavuudeksi luokitellaan sellainen havaintojen osajoukko, jotka eivät yksittäisinä datapisteinä ole poikkeavia, mutta joukkona poikkeavat muusta datasta (Chandola et al. 2009). Kollektiivisia poikkeavuuksia esiintyy tyypillisimmin aikasarjamuotoisissa aineistoissa, kun esimerkiksi joukko havaintoja saa tyypillisiä arvoja mutta epätyypillisenä ajanhetkenä.

3.2 Koneoppiminen

Koneoppiminen on tietojenkäsittelytieteen osa-alue, joka yleisesti liitetään osaksi tekoälyä ja sen sovelluksia. Koneoppimismenetelmiä on kehitetty jo vuosikymmeniä, mutta vasta toisen digitalisaatioaallon myötä koneoppimismenetelmiä hyödyntäviä sovelluksia on alettu kehittämään suuressa mittakaavassa. Tehokkaiden tietokoneiden sekä niin ikään uudenlaisen datan, niin kutsutun Big Datan, myötä koneoppimismenetelmien käyttöönotto on ollut ensinnäkin mahdollista mutta myös tarpeellista.

Koneoppimismenetelmiä on perinteisesti jaoteltu kahteen pääluokkaan oppimisproseduuriensa perusteella, jotka ovat ohjattu (supervised) ja ohjaamaton (unsupervised) oppiminen. Näiden kahden pääluokan lisäksi puhutaan myös osittain ohjatusta oppimisesta (semi-supervised learning) sekä vahvistusoppimisesta (reinforcement learning). Pääluokkien alla menetelmiä voidaan jaotella edelleen instanssi- ja malliperusteisiin menetelmiin. Instanssiperusteisia menetelmiä kutsutaan myös laiskoiksi oppijoiksi (lazy learner) ja mallipohjaisia ahneiksi oppijoiksi (eager learning).

3.2.1 Ohjattu oppiminen

Ohjatun oppimisen menetelmissä lähtökohtana on, että opetusaineisto sisältää algoritmin suorittamaan tehtävään oikeat vastaukset. Esimerkiksi luokittelualgoritmien kohdalla opetusaineistossa on luokkamuuttuja, joka sisältää jokaisen datapisteen oikean luokan. Algoritmin tehtävänä on oppia opetusvaiheessa sellainen tavoitefunktio $f: X \rightarrow Y$, joka parhaiten projisoi lähtöjoukon X instanssit x_i maalijoukon Y arvoiksi y_j. Luokittelualgoritmeissa maalijoukko Y on diskreetti muuttuja, kun taas regressiopohjaisissa menetelmissä vastemuuttuja on jatkuva. Käytetyimpiä ohjatun oppimisen algoritmeja ovat muun muassa erilaiset päätöspuu- ja -metsämenetelmät, neuroverkot, tukivektorikoneet sekä erilaiset regressiomenetelmät. Ohjatun oppimisen menetelmissä on tärkeää ottaa huomioon mallin kompleksisuus verrattuna käytettävissä olevaan aineistoon. Mitä enemmän dataa on käytettävissä, sitä kompleksisempi malli voidaan valita. Mikäli käytössä oleva aineisto on pieni tai muuttujat ovat epätasapainoisia, voi liian kompleksinen malli johtaa ylisovittamiseen. Ylisovittava malli oppii opetusaineiston tarkasti, mutta yleistys testi- ja validointiaineistoon on usein heikkoa. (Kotsiantis, 2007)

3.2.2 Ohjaamaton oppiminen

Toisin kuin ohjatun oppimisen menetelmissä, ohjaamattomassa oppimisessa ei ole ennakkotietoa ennustettavan muuttujan oikeista arvoista. Ohjaamattoman oppimisen menetelmien tavoitteena on muodostaa opetusaineiston avulla malli, joka tunnistaa datasta säännönmukaisuuksia ja näiden avulla osaa esimerkiksi ryhmitellä samankaltaisimmat datapisteet samoihin ryhmiin. Ohjaamattoman oppimisen algoritmeja käytetään erityisesti aineistossa olevien säännönmukaisuuksien ja muuttujien välisten suhteiden kuvaamiseen, kuten klusterointiin sekä dimensioiden redusointiin ennemmin kuin varsinaiseen vastemuuttujan ennustamiseen. (Klawonna, 2016)

Klusterointimenetelmät voidaan jakaa osittaviin, hierarkkisiin sekä mallipohjaisiin menetelmiin. Osittavat ja hierarkkiset menetelmät ovat instanssiperusteisia menetelmiä tarkoittaen, että instanssien välisiä suhteita mitataan ilman oletuksia niiden jakaumista esimerkiksi etäisyyden tai esiintymistiheyden perusteella. Suosittujen klusterointimenetelmien joukkoon kuuluvat muun muassa K-means-klusteroinnin eri muunnelmat, jotka perustuvat datapisteiden välisten etäisyyksien mittaamiseen sekä DBSCAN, joka taas perustuu datapisteiden alueittaisiin esiintymistiheyksiin.

3.2.3 Osittain ohjattu oppiminen

Osittain ohjattu oppiminen on ohjatun ja ohjaamattoman oppimisen välimuoto ja sitä hyödynnetään tilanteissa, jolloin opetusaineisto sisältää sekä datapisteitä oikeilla vastemuuttujan arvoilla tai luokitteluilla että datapisteitä, joiden vastemuuttujan oikea arvo ei ole tiedossa. Osittain ohjatun oppimisen menetelmissä opetusaineisto jaetaan usein kahteen ryhmään siten, että toisen ryhmän havainnoilla vastemuuttujan oikeat arvot tai luokat ovat tiedossa ja toisen ryhmän datapisteillä ei. Osittain ohjattua oppimista käytetään esimerkiksi tilanteissa, joissa ohjaamattomalla oppimisella kerätty esitieto muuttujien jakaumista auttaa tarkentamaan ohjatun oppimisen tuloksia, kun opetusaineiston luokitellut datapisteet mallinnetaan ohjatun oppimisen menetelmää käyttäen. (Chapelle, Schölkopf & Zien, 2006)

3.2.4 Vahvistusoppiminen

Vahvistusoppimisen menetelmät perustuvat ympäristöstä saatuun palautteeseen. Menetelmä, kutsuttakoon tässä oppijaksi, pyrkii maksimoimaan ympäristöstä saadun positiivisen palautteen. Kuten ohjaamattomassa oppimisessa, vastemuuttujan oikeat arvot eivät ole läsnä opetusaineistossa. Vahvistusoppiminen kuitenkin

eroaa olennaisesti ohjaamattomasta oppimisesta siinä, ettei tavoitteena ole löytää säännönmukaisuuksia datan rakenteesta vaan yksinomaan maksimoida saatu positiivinen palaute suorituksista. (Sutton & Barto, 2017)

3.3 Koneoppiminen

Tutkielmassa keskitytään ohjatun ja ohjaamattoman oppimisen menetelmiin. Klusterointimenetelmillä viitataan ohjaamattomaan oppimiseen, joita ensisijaisesti käytetään apuna hahmottamaan aineiston rakennetta sekä muuttujien välisiä suhteita. Luokittelumenetelmillä viitataan ohjatun oppimisen menetelmiin, jotka hyödyntävät opetusaineiston sisältämää tietoa havaintojen oikeasta luokasta. Tutkielmassa poikkeavien havaintojen tunnistamista käsitellään kaksiluokkaisena luokitteluongelmana, jossa ennustettavalla luokalla on kaksi mahdollista arvoa.

3.3.1 K-means

Yksi käytetyimmistä koneoppimisalgoritmeista on K-means-algoritmi, joka on perinteinen optimointimenetelmien luokkaan kuuluva klusterointialgoritmi. Optimointimenetelmien tavoitteena on joko minimoida tai maksimoida tavoitefunktio iterointien kautta. K-means-algoritmin tapauksessa tavoite on minimoida klustereiden sisäinen varianssi. Optimoitava tavoitefunktio on jäännösneliösumma (1), joka määrittää klustereiden keskipisteiden ja klustereissa sijaitsevien alkioiden välisen euklidisen etäisyyden neliöiden summan. Jäännösneliösumma (SSE) ja euklidinen etäisyys (d_e) ovat muotoa

$$SSE = \sum_{i=1}^{N} d_e\left(x_i, c_j\right)^2, \text{missä} \qquad (1)$$

$N = Alkioiden\ lukumäärä,$
$x_i = Datajoukon\ alkio,$
$c_j = \frac{\sum x_i}{N_j} = \bar{x}_j\ (Alkion\ x_i\ sisältävän\ klusterin\ j\ keskipiste),$
$d_e = \sqrt{(x_i - c_j)^2}$

K-means-algoritmin ensimmäiset käyttösovellukset esitettiin jo vuonna 1967 James MacQueenin toimesta. MacQueen esitti menetelmän käyttösovelluksiksi muun muassa samankaltaisuuden ryhmittelyn (klusteroinnin), monimuuttujaisten jakaumien arvioinnin sekä parametrittoman riippumattomuuden testauksen useiden muuttujien tapauksessa. Suosituimmaksi menetelmän käyttösovellukseksi on osoittautunut klusterointi, jota nykyisin käytetään monien uusien, monimutkaisempien menetelmien perustana. MacQueenin esittämä algoritmi ottaa syötteenään kolme parametria: klustereiden aloituslukumäärän K, klustereiden keskipisteiden välisen minimietäisyyden C sekä alkion ja klusterin keskipisteen välisen maksimietäisyyden R. Algoritmin ensimmäisessä vaiheessa asetetaan K ensimmäistä datajoukon alkiota edustamaan klustereiden keskipisteitä, jonka jälkeen jokaisen alkion etäisyys klustereiden keskipisteisiin lasketaan ja alkio asetetaan klusteriin,

jonka keskipisteen etäisyys alkiosta on pienin. Tämän jälkeen lasketaan uusi keskiarvo klusterille, johon alkio lisättiin. Mikäli alkion etäisyys lähimmän klusterin keskipisteestä on suurempi kuin määrätty maksimietäisyys R, muodostetaan alkiosta uusi klusteri. Kunkin (ennalta olleeseen klusteriin) lisätyn alkion ja uuden keskiarvon määrittämisen jälkeen lasketaan kahden toisiaan lähimpänä olevan klusterin keskipisteiden välinen etäisyys. Jos tämä etäisyys on pienempi kuin määrätty minimietäisyys C, yhdistetään klusterit. Näin jatketaan, kunnes kaikkien klustereiden välinen etäisyys ≥ C. Näin ollen klustereiden lukumäärä vaihtelee prosessin aikana, eikä lopullista klustereiden lukumäärää tiedetä ennalta. Kun kaikki alkiot on sijoitettu klustereihin, aloitetaan kierros alusta käyttäen ensimmäisellä kierroksella muodostettujen klustereiden keskipisteitä uuden kierroksen keskipisteiden lähtöarvoina. (MacQueen, 1967) Algoritmi 1 esittää alkuperäisen menetelmän toimintaperiaatteen.

Algoritmi 1: *MacQueen(D, R, C, K)*

Syöte: *Alkiot sisältävä aineisto D, maksimietäisyys R, minimietäisyys C, klustereiden aloituslukumäärä K*
Tuloste: *M klusteria, joiden välinen etäisyys ≥ C*

1. *Aseta K ensimmäistä alkiota edustamaan klustereiden keskipisteitä*
2. *Jokaiselle alkiolle $x_i \in D$ laske etäisyys d_{ij} jokaisen klusterin j keskipisteeseen c_j*
   a. *Jos ($min(d_{ij}) \geq R$)*
      Aseta x_i uuden klusterin j = k +1 keskipisteeksi
   b. *Muutoin, jos ($min(d_{ij}) \geq C$)*
      Aseta x_i klusteriin j ja laske klusterin uusi keskipiste c_j
   c. *Muutoin*
      Yhdistä klusterit, joiden välinen etäisyys ≤ C
3. *Toista kohtia 1. ja 2., kunnes klusterit ja niiden keskipisteet eivät enää muutu alkioita siirtämällä*

Pysähdy

Vuosikymmenten saatossa algoritmi on hieman muuttanut muotoaan ja yksi nykyisin tunnetuimmista ja käytetyimmistä K-means-algoritmeista pohjautuu Stuart P. Lloydin vuonna 1982 julkaisemaan menetelmään, jonka Lloyd kehitti pulssikoodimodulaatioon. Lloydin algoritmi on MacQueenin algoritmia yksinkertaisempi, sillä minimi- ja maksimietäisyydet eivät ole määrättyjä, eikä kyseisiä etäisyysvertailuja suoriteta. Näin ollen Lloydin algoritmi on aikavaatimukseltaan MacQueenin algoritmia tehokkaampi suorittaessaan ainoastaan alkion ja klustereiden keskipisteiden välisen etäisyyden laskennan, jonka jälkeen alkio asetetaan aina klusteriin, jonka keskipisteeseen etäisyys on pienin. Tämän jälkeen klusterille lasketaan uusi keskipiste. Iterointikierroksia suoritetaan, kunnes klustereiden keskipisteet eivät enää muutu. (Lloyd, 1982)

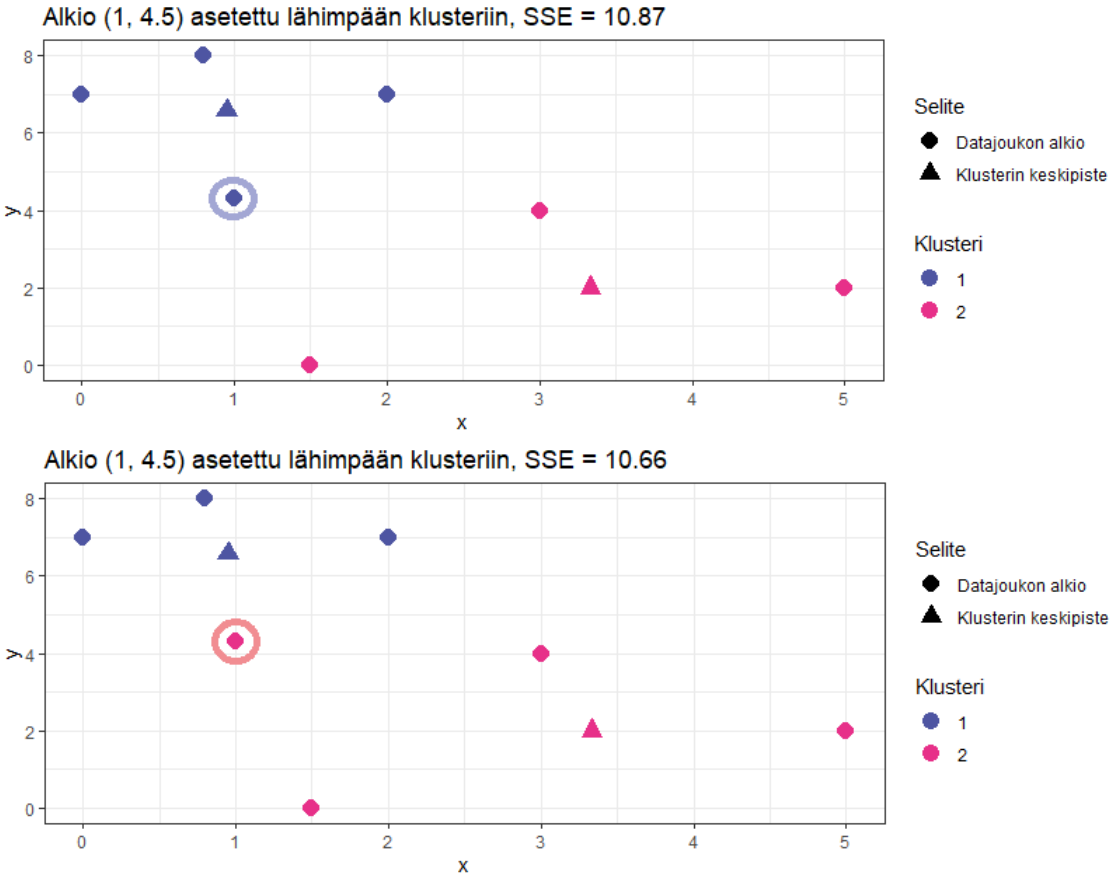
Algoritmi 2: *Lloyd(D, K)*

Syöte: *Alkiot sisältävä aineisto D, klustereiden lukumäärä K*
Tuloste: *K klusteria*

1. *Aseta K mielivaltaisesti valittua alkiota edustamaan klustereiden keskipisteitä*
2. *Jokaiselle alkiolle $x_i \in D$*
   Laske etäisyys d_{ij} jokaisen klusterin j keskipisteeseen c_j, j = 1, …, K
   Aseta x_i klusteriin j, jonka keskipisteeseen c_j etäisyys d_{ij} on pienin
3. *Laske klusterin j uusi keskipiste*
4. *Toista kohtia 2. ja 3. kunnes klustereiden keskipisteet eivät enää muutu*

Pysähdy

K-means-klusterointiin on kehitetty myös muita algoritmeja, kuten Elkanin algoritmi lyhentämään prosessin suoritusaikaa sekä Hartigan-Wong –algoritmi tehostamaan optimoinnin tarkkuutta. Hartigan-Wong –algoritmi pyrkii päätymään lokaaliin optimiratkaisuun, jossa koko klusteriavaruuden jäännösneliösumma on minimoitu. Algoritmi laskee jokaiselle alkiolle etäisyyden sekä lähimpään että toiseksi lähimpään klusterikeskipisteeseen. Ensin alkiot sijoitetaan klustereihin, joiden keskipiste on lähimpänä alkiota. Tämän jälkeen lasketaan, pieneneekö jäännösneliösumma, mikäli alkio siirretään klusteriin, jonka keskusta on seuraavaksi lähimpänä. Jos jäännösneliösumma pienenee, siirretään alkio tähän toiseen klusteriin. Tällöin yksittäiset alkiot saattavat kuulua klusteriin, jonka keskipiste ei ole kaikkein lähimpänä, mikäli alkion kuuluminen tähän klusteriin pienentää kokonaisjäännösneliösummaa (kuva 1). (Hartigan & Wong, 1979)



Kuva 1 Hartigan-Wong algoritmi ei aina aseta alkioita lähimpiin klustereihinsa

---

Algoritmi 3: *Hartigan-Wong(D, K)*

Syöte: *Alkiot sisältävä aineisto D, klustereiden lukumäärä K*

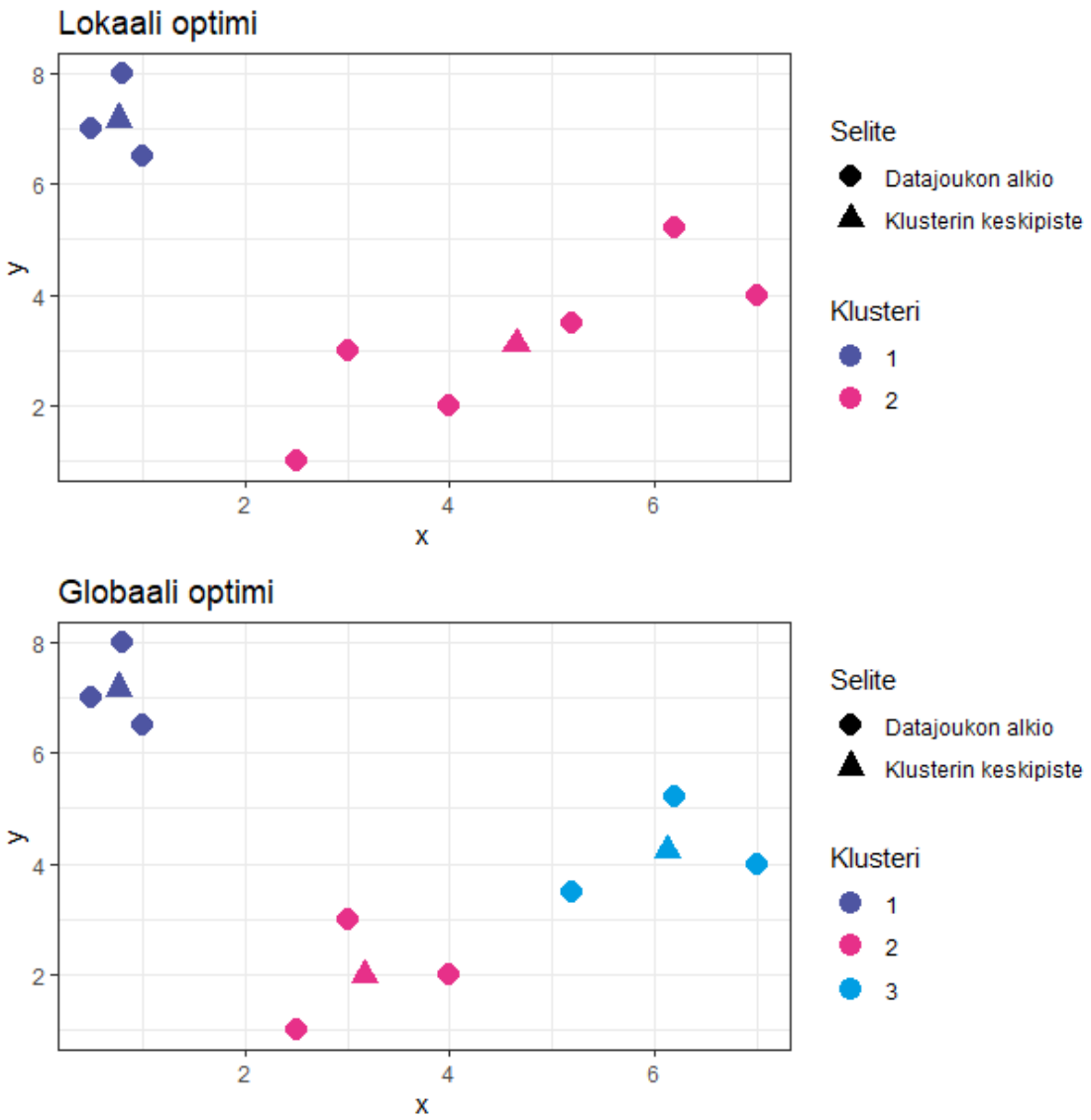
Tuloste: *K klusteria*

1. *Aseta K mielivaltaisesti valittua alkiota edustamaan klustereiden keskipisteitä*
2. *Jokaiselle alkiolle $x_i \in D$*
   Laske etäisyys d_{ij} jokaisen klusterin j keskipisteeseen c_j, j = 1, …, K
   Aseta x_i klusteriin j, jonka keskipisteeseen etäisyys d_{ij} on pienin
3. *Laske klusterin j uusi keskipiste c_j*
4. *Jokaiselle klusterille j*
   a. *Laske klusterin sisäinen keskineliövirhe SSE_j*
   b. *Jokaiselle alkiolle x_i klusterissa j*
      Laske vaihtoehtoinen SSE_k kaikille klustereille k≠j, jos alkio x_i sijoitettaisiin klusteriin k
      Jos $SSE_k < SSE_j$
         Siirrä alkio x_i klusterista j klusteriin k
5. *Toista kohtia 3. ja 4., kunnes jäännösneliösumma SSE ei enää pienene alkioita siirtämällä*

Pysähdy

MacQueen esitti tutkimuksessaan kaksi merkittävintä K-means-menetelmän varjopuolta, jotka osittain on onnistuttu ratkaisemaan menetelmän myöhemmissä algoritmeissa. MacQueen huomasi, että kun aineiston havaintojen järjestystä muutetaan, tuloksena saadut klusterit muuttuivat. MacQueen kuitenkin katsoi tämän marginaaliseksi ongelmaksi ja totesi, että pääasiassa eroavaisuudet tuloksissa johtuivat alkioista, jotka sijoittuivat kahden klusterin väliin. Tällöin toistettaessa prosessia nämä alkiot luokiteltiin vaihtelevasti jompaankumpaan lähimmistä klustereista. Myöhemmin on kuitenkin todettu, että lähtökeskipisteet voivat vaikuttaa merkittävästikin klusteroinnin tuloksiin.

Toinen, merkittävämpi K-means-menetelmän ongelma, jonka jo MacQueen havaitsi, on taipumus päätyä optimoinnissa lokaaliin minimiin globaalin ratkaisun sijaan (kuva 2). Lokaaliin optimiin päätymisen ongelmaan on ehdotettu ratkaisuksi muun muassa iterointikierrosten kasvattamista niin suureksi, että löydetään kaikki mahdolliset lokaalit optimit ja valitaan näistä paras, jonka on oltava globaali ratkaisu. On kuitenkin havaittu, että lokaaleita optimeja voi olla jopa tuhansia, joten kaikkien ratkaisuiden etsintä ei ole mielekästä suoritusajan kannalta. (MacQueen, 1967) Kahden edellä mainitun lisäksi menetelmään liittyy alttius tyypin I poikkeaville havainnoille, joka on seurausta menetelmän matemaattisista ominaisuuksista. Tyypin I poikkeavuudet eli virheelliset ääriarvot aiheuttavat harhaa keskiarvon laskentaan ja näin ollen vaikuttavat virheellisesti keskiarvojen laskentaan perustuvien menetelmien tuloksiin. Menetelmästä on kehitelty erinäisiä muunnelmia näiden varjopuolten korjaamiseksi, joista muutama esitellään tulevissa luvuissa.



Kuva 2 K-means -menetelmällä on tendenssi päätyä lokaaliin optimiin globaalin optimin sijaan

3.3.2 K-means++

Kuten jo MacQueen havaitsi, K-means-menetelmän tulokset ovat riippuvaisia keskipisteistä, jotka klusteroinnin alkuarvoiksi valitaan. Alun perin klustereiden ensimmäiset keskipisteet valittiin joko ottamalla datajoukon K ensimmäistä alkiota tai valitsemalla satunnaisesti K alkiota datajoukosta. K-means-menetelmän aloituskeskipisteet muodostuvat sattuman kaupalla, joten alkuarvot voivat olla hyvinkin sopimattomia, jos esimerkiksi K:n klusterin keskipisteeksi valikoituu ainoastaan toisiaan lähellä olevia pisteitä.

David Arthur sekä Sergei Vassilvitskii esittivät vuonna 2007 vaihtoehtoisen tavan klustereiden aloituskeskipisteiden valintaan, jonka he olivat todenneet lisäävän sekä algoritmin suoritusnopeutta että -tarkkuutta. Parivaljakko esitti, että valitsemalla vain ensimmäinen keskipiste satunnaisesti ja tämän jälkeen asettamalla seuraavaksi keskipisteeksi alkio, jonka todennäköisyys seuraavaksi keskipisteeksi on suurin kaavan 3 mukai-

sesti laskettuna, algoritmin suoriutumiskyky paranee. Määrittämällä kullekin datajoukon alkiolle todennäköisyys edustaa seuraavan klusterin keskipistettä pyritään välttämään tilanne, jossa klustereiden keskipisteiksi valitut alkiot olisivat liian lähekkäin toisiaan. Toisaalta pyritään myös välttämään ongelma, joka mahdollisesti ilmenisi valitsemalla aina klusterin keskipisteestä kauimmainen alkio seuraavan klusterin keskipisteeksi; tällöin ääriarvot päätyisivät todennäköisimmin edustamaan ensimmäisten klustereiden keskipisteitä. Todennäköisyys, että piste $x_i \in D$ valitaan seuraavan klusterin j+1 keskipisteeksi c_{j+1}, on muotoa

$$p(x_i = c_{j+1}) = \frac{d_e(x_i, c_j)^2}{\sum_{x \in D} d_e(x, c_{j+1})^2}, missä \qquad (3)$$

$c_j = Edellisen\ asetetun\ klusterin\ j\ keskipiste\ \bar{x}_j,$
$D = Datajoukko\ ja$
$d_e = Euklidinen\ etäisyys$

(Arthur & Vassilvitskii, 2007)

Algoritmi 4: *K-means++(D, K)*

Syöte: *Alkiot sisältävä aineisto D, Klustereiden lukumäärä K*
Tuloste: *K klusteria*

1. *Valitse satunnainen alkio $x_i \in D$ ensimmäisen klusterin j keskipisteeksi*
2. *Kunnes K klusteria on valittu*
   Valitse uuden klusterin j keskipisteeksi $x_i \in D$, jonka todennäköisyys $p(x_i=c_j)$ on suurin
3. *Laske klusterin j uusi keskipiste*
4. *Toista kohtia 2. ja 3. kunnes klustereiden sisäinen jäännösneliösumma SSE ei pienene alkioita siirtämällä*
5. *Palauta k klusteria*

Pysähdy

3.3.3 K-medoids

Yksi K-means-algoritmin heikkouksista on herkkyys poikkeaville havainnoille. Koska algoritmin optimointi perustuu alkioiden etäisyyksien keskiarvojen laskentaan klustereiden sisällä, saavat ääriarvon omaavat alkiot tarpeettoman suuren painon optimoinnissa. Tämän ominaisuuden poistamiseksi L. Kauffman yhdessä P. J. Rousseeuw:n kanssa kehittivät vaihtoehtoisen menetelmän K-means:lle, joka keskiarvojen laskennan sijaan perustuu alkioiden erilaisuuden mittaamiselle. Tämä menetelmä tunnetaan K-medoids-nimellä, ja sen kuuluisin sovellusalgoritmi on Kauffmanin ja Rousseeuw:n esittelemä PAM (Partitioning Around Methods). PAM-algoritmi toimii parhaiten pienten aineistojen tapauksessa, mutta samaisessa julkaisussa esiteltiin myös vaihtoehtoinen algoritmi CLARA suurempien aineistojen klusterointiin.

Toisin kuin K-means-klustereiden keskipisteitä laskettaessa, K-medoid-menetelmässä ei muodosteta lainkaan keinotekoisia klustereiden keskipisteitä. Sen sijaan valitaan datajoukon alkioista K medoidia kullekin klusterille. Algoritmi koostuu kahdesta vaiheesta; rakennus- ja vaihtovaiheesta. Rakennusvaiheessa algoritmi

etsii sopivimmat K alkiota $x_i \in X$ edustamaan K:n klusterin medoidia ja sijoittaa kaikki datajoukon alkiot klustereihin siten, että erilaisuus klusterin medoidiin minimoituu. Erilaisuus lasketaan jonkin etäisyysmitan, kuten euklidisen etäisyyden, mukaan. Rakennusvaiheen ensimmäiseksi medoidiksi valitaan koko datajoukon keskimmäisin alkio eli alkio, jonka erilaisuus muihin datajoukon alkioihin verrattuna on pienin. Tämän jälkeen lasketaan kunkin valitsemattoman alkion x_j erilaisuus D_j lähimpään medoidiin sekä erilaisuus d(j,i) seuraavaksi medoidiksi harkittavaan alkioon x_i. Mikäli näiden erilaisuuksien erotus on nollaa suurempi, vastaa se x_j:*n* kontribuutiota (kaava 4) valita alkio x_i seuraavaksi medoidiksi. Jokaiselle alkiolle x_i lasketaan kontribuutioiden summa ja seuraavaksi medoidiksi valitaan alkio x_i, joka maksimoi kontribuutioiden summan. Tämä vaihe toistetaan, kunnes K medoidia on valittu. (Kauffman & Rousseeuw, 1990) Kontribuutio lasketaan kaavalla

$$\sum_j C_{ij} = \max \sum \big(D_j - d(j,i), 0\big), missä \qquad (4)$$

D_j = *Alkion* x_j *erilaisuus lähimpään medoidiin ja*
$d(j,i)$ = *Alkion* x_j *erilaisuus alkioon* x_i
Erilaisuuden mittana voidaan käyttää esimerkiksi euklidista etäisyyttä.
(Kauffman & Rousseeuw, 1990, s.103-104)

Kun K medoidia on valittu, kukin alkio $x_i \in X$ sijoitetaan klusteriin, jonka medoidiin erilaisuus on pienin. Tämän jälkeen algoritmi siirtyy vaihtovaiheeseen. Vaihtovaiheessa kunkin medoidin m_i ja medoidiksi valitsemattoman alkion x_h välillä harkitaan vaihtoa siten, että x_h vaihdetaan medoidiksi. Vaihdon kustannukseksi lasketaan toisen valitsemattoman alkion x_j kontribuutio vaihtoon (kaava 5). Mikäli kontribuutioiden yhteenlaskettu summa on negatiivinen, vaihto suoritetaan. Muussa tapauksessa todetaan, ettei vaihto vähennä erilaisuutta ja siirrytään seuraavaan medoidi-alkio −pariin (m_i,x_h). Kontribuution laskenta voidaan esittää yhtälöryhmänä

$$C_{jih} = \begin{cases} \mathbf{0}, jos\ d(j,i) > d(j,m)\ ja\ d(j,h) > d(j,m) \\ \boldsymbol{d(j,h)} - \boldsymbol{d(j,i)},\ jos\ d(j,h) < E_j \\ \boldsymbol{E_j} - \boldsymbol{D_j},\ jos\ d(j,h) \geq E_j \end{cases} , missä \qquad (5)$$

$Medoidi\ m \neq m_i$,
E_j = *erilaisuus* x_j*:n ja toiseksi lähimmän medoidin välillä ja*
D_j = x_j*:n etäisyys lähimpään medoidiin*

Vaihtoehtoiskustannus saadaan kontribuutioiden summana kaavan 6 mukaisesti.

$$T_{ih} = \sum_j C_{jih} \qquad (6)$$

Algoritmin vaihtovaihe päättyy, kun enempää vaihtoja ei voida tehdä siten, että kokonaiserilaisuus pienenisi. Algoritmi 5 esittää K-medoids-menetelmän toimintaperiaatteen.

Algoritmi 5: *K-medoids(X, K)*

Syöte: *Alkiot sisältävä aineisto X, Klustereiden lukumäärä K*

Tuloste: *K klusteria*

1. *Rakennusvaihe*
    a. *Valitse datajoukon keskimmäinen alkio medoidiksi m_1*
    b. *Kunnes K medoidia on valittu*
        Jokaiselle valitsemattomalle alkiolle $x_i \in X$
            Laske valitsemattomien alkioden $x_j \in X$ kontribuutio C_{ij}
        Aseta medoidiksi m_i alkio x_i, joka maksimoi kontribuutioiden summan
    c. *Aseta valitsemattomat alkiot $x \in X$ klustereihin lähimmän medoidin mukaan*
2. *Vaihtovaihe*
    Jokaiselle parille (m_i, x_j), jossa x_j on klusterin j medoidi

        Laske vaihtokustannus T_{ih}

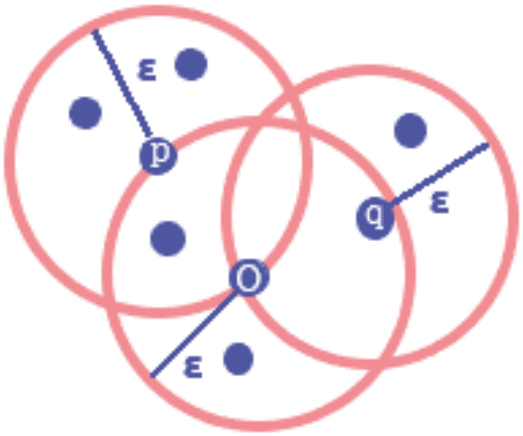
        Jos $T_{ih} < 0$
            Tee vaihto $x_j \rightarrow m_i$
3. *Toista kohtaa 2 kunnes $T_{ih} \geq 0$ kaikille $m_i \in M$, $x_h \in X$*

Pysähdy

3.3.4 DBSCAN

DBSCAN (Density Based Spatial Clustering of Applications with Noise) on ohjaamattoman oppimisen klusterointimenetelmä, jonka Ester, Kriegel, Sandel ja Xu (1996) kehittivät erityisesti spatiaalisille aineistoille. Työssään Ester et al. osoittavat, että algoritmi pystyy paitsi klusteroimaan datan tarkasti vähällä etukäteistiedolla aineiston rakenteesta, pystyy myös muodostamaan muodoltaan mielivaltaisia klustereita sekä on luokittelutarkkuudeltaan hyvä myös huomattavan suurien aineistojen tapauksessa. DBSCAN-klusterointi perustuu datapisteiden esiintymistiheyteen. Klusterit muodostuvat alueista, joilla datapisteiden esiintymistiheys on suuri verrattuna klusteria ympäröiviin alueisiin. Tällöin klustereiden ulkopuolisilla, harvoilla, alueilla sijaitsevat datapisteet voidaan luokitella poikkeuksellisiksi ja näin ollen menetelmä soveltuu hyvin poikkeavuuksien tunnistukseen.

DBSCAN-menetelmässä kukin datapiste luokitellaan joko ydin- tai reunapisteeksi tai poikkeavuudeksi. Luokittelu tehdään syöteparametrien ε-naapurusto sekä minPts pohjalta. ε-naapurusto kuvaa naapuruston säteen pituutta ja minPts naapurustossa sijaitsevien naapureiden vähimmäislukumäärää. Datapiste luokitellaan ydinpisteeksi, mikäli sen ε-naapurustossa sijaitsee vähintään minPts datapistettä. Reunapisteiksi taas luokitellaan pisteet, jotka eivät ole ydinpisteitä, mutta sijaitsevat ydinpisteen naapurustossa. Klustereiden määrittelyyn kuuluu edellä mainittujen lisäksi kaksi käsitettä, tiheys-saavutettavuus sekä tiheys-liitännäisyys. Kaksi pistettä ovat tiheys-saavutettavia toisiinsa nähden silloin, jos niiden välille voidaan muodostaa sellainen pisteiden ketju, jossa jokainen piste kuuluu edellisen pisteen ε-naapurustoon ja edellinen piste on ydinpiste. Tiheys-liitännäisyys taas pätee kahden pisteen välillä silloin, jos on olemassa p, jonka kanssa molemmat pisteet ovat tiheys-saavutettavia kuvan 3 mukaisesti. Klusteri määritellään tiheys-liitännäisyyden avulla siten, että klusterin muodostaa suurin mahdollinen lukumäärä sellaisia datapisteitä, jotka ovat joko ydin- tai reunapisteitä ja ovat tiheys-liitännäisiä toisiinsa nähden. Poikkeavuuksiksi määritellään yksinkertaisesti sellaiset datapisteet, jotka eivät kuulu yhteenkään klusteriin. (Ester et al., 1996)



Kuva 3 Pisteet p ja q ovat tiheys-liitännäisiä toisiinsa nähden pisteen o kautta

Klustereiden muodostus alkaa mielivaltaisesti valitusta lähtöpisteestä p, jolle etsitään annetuilla epsilon- ja minPts-arvoilla kaikki tiheys-liitännäiset pisteet. Mikäli tiheys-liitännäisiä pisteitä löytyy, luokitellaan piste p ydinpisteeksi ja muodostetaan klusteri. Mikäli tiheys-liitännäisiä pisteitä ei löydetä, siirrytään seuraavaan pisteeseen. Algoritmi etenee toistaen etsinnän erikseen jokaiselle datajoukon pisteelle. DBSCAN-algoritmin toimintaperiaate on kuvattu algoritmissa 6.

Algoritmi 6: *DBSCAN(D, ε, minPts)*

Syöte: *Aineisto D, naapuruston säde ε, naapureiden vähimmäislukumäärä minPts*

Tuloste: C klusteria

C = 0
Jokaiselle pisteelle pϵD

1. *Merkitse piste käydyksi*
2. *Laske naapureiden lukumäärä säteellä ε*
3. *Jos naapureiden lukumäärä ≥ minPts*
    a. *Merkitse p ydinpisteeksi*
    b. *Jos ε-naapurustosta löytyy ydinpiste*
       Aseta p klusteriin C+1
    c. *Muutoin*
       Aseta p klusteriin C
4. *Muutoin, jos ε-naapurustosta löytyy ydinpiste*
    a. *Merkitse p reunapisteeksi*
    b. *Aseta p klusteriin C*
5. *Muutoin*
   Merkitse p poikkeavuudeksi
6. *Palauta C klusteria*

3.3.5 K:n lähimmän naapurin menetelmä

Yksi suosituimmista ohjatun oppimisen luokittelumenetelmistä on instanssiperusteinen k:n lähimmän naapurin menetelmä (k-Nearest Neighbours, k-NN). Menetelmän suosio perustuu sen helppoon tulkittavuuteen sekä alhaiseen laskentavaativuuteen.

K:n lähimmän naapurin toimintaperiaate pohjautuu instanssien välisiin etäisyyksiin. Jokaisen instanssin oletetaan olevan piste m-dimensionaalisessa avaruudessa, jossa m on datajoukon dimensioiden lukumäärä ja dimensioista yksi edustaa instanssien luokkaa. Instanssit pyritään luokittelemaan siten, että lähellä toisiaan sijaitsevat instanssit edustavat samaa luokkaa. Luokittelija laskee luokiteltavan instanssin etäisyyden tämän k:hon lähimpään naapuriin. Instanssi luokitellaan samaan luokkaan lähimmän naapurinsa kanssa, kun k=1. Mikäli k>1, määräytyy instanssin luokka yksinkertaisimmillaan sen mukaan, mihin luokkaan suurin osa naapureista kuuluu. K:n arvo asetetaan välille [1,n], kun n on instanssien lukumäärä.

ETÄISYYS

Instanssin etäisyys naapureihinsa lasketaan jokaisen muuttujan suhteen. Homogeenisille aineistoille etäisyysmittoja on lukuisia erilaisia, joista perinteisin on euklidinen etäisyys. Chomboon et al. kartoittivat tutkimuksessaan yhdentoista eri etäisyysmitan tehokkuutta k-NN-menetelmää käytettäessä. Chomboon et al. löysivät euklidisen etäisyyden lisäksi viisi muuta etäisyysmittaa, jotka antoivat tarkkoja tuloksia k:n lähimmän naapurin luokittelutehtävässä; Mahalanobis-, Manhattan-, Chebychev- sekä standardoitu euklidinen etäisyys. Kaikki näistä etäisyysmitoista ovat homogeenisten aineistojen tapauksessa käytettäviä. (Chomboon, Chujai, Teerarassamee, K. Kerdprasop & N. Kerdprasop, 2015)

Manhattan- ja euklidinen etäisyys ovat niin kutsutun Minkowski-etäisyyden (7) erityistapauksia, joissa p=1 Manhattan-etäisyydelle ja p=2 euklidiselle etäisyydelle. Kolmas Minkowski-etäisyyden erityistapaus, jonka Chomboon et al. totesivat olevan käyttökelpoinen lähimmän naapurin menetelmässä, on Chebyshev-etäisyys (8), jossa p=∞. Minkowski-etäisyyttä sovellettaessa muuttujat tulee skaalata samalle asteikolle esimerkiksi normalisoimalla arvot tietylle välille. Minkowski-etäisyys lasketaan datapisteiden ja niiden naapureiden etäisyyksien summana, ja on muotoa

$$d_{Minkowski} = (\sum_{i=1}^{a} |x_i - y_i|^p)^{\frac{1}{p}} \text{ , missä} \qquad (7)$$

$a = Muuttujien\ lukum$äärä,
$x_i = Muuttujan\ i\ arvo\ instanssilla\ x\ ja$
$y_i = Muuttujan\ i\ arvo\ x{:}n\ naapurilla\ y$

$$d_{Chebyshev} = \lim_{p\to\infty}(\sum_{i=1}^{a}|x_i - y_i|^p)^{\frac{1}{p}} = \max_i(|x_i - y_i|) \text{ , missä} \quad (8)$$

$x_i = Muuttujan\ i\ arvo\ instanssilla\ x\ ja$
$y_i = Muuttujan\ i\ arvo\ x{:}n\ naapurilla\ y$

Mahalanobis-etäisyys (9) on pisteen ja jakauman välistä etäisyyttä mittaava suure, jota voidaan soveltaa monimuuttujaisessa aineistossa. Mahalanobis-etäisyyden hyöty verrattuna Minkowski-etäisyyteen on, että se huomioi myös aineiston kovarianssirakenteen. Tällöin, mikäli muuttujat korreloivat keskenään, tulee muuttujien väliset korrelaatiot huomioitua etäisyyksien mittaamisessa. Mahalanobis-etäisyys on muotoa

$$d_{Mahalanobis} = \sum_{i=1}^{a}\sqrt{\frac{(x_i-\bar{x})^2}{\sigma^2}} \text{ , missä} \quad (9)$$

$a = Muuttujien\ lukumäärä,$
$\bar{x} = Muuttujan\ i\ keskiarvo\ ja$
$\sigma^2 = Muuttujan\ sisäinen\ varianssi$

(Chomboon et al., 2015)

Yllä esitellyt etäisyysmitat soveltuvat ainoastaan kvantitatiivisten muuttujien etäisyyksien mittaamiseen. Näin ollen sekatyyppisten aineistojen kohdalla täytyy aineiston nominaaliset muuttujat joko muuntaa numeerisiksi esimerkiksi *one-hot*-koodauksella tai käyttää etäisyyksien mittaamiseen suureita, jotka pystyvät käsittelemään myös nominaalisia muuttujia. *One-hot*-koodauksessa kategorisista muuttujista muodostetaan kullekin muuttujan arvolle uusi binäärinen muuttuja. *One-hot* -koodatut muuttujat saavat arvon 1 silloin, kun alkuperäinen muuttuja edustaa kyseistä luokkaa ja arvon 0 muulloin. Wilson ja Martinez (1997) esittelivät heterogeenisissa aineistossa etäisyyden määrittämiseen useita funktioita, joista nykyisin suosituimpiin kuuluu HEOM-funktio (Heterogeneous Euclidean-Overlap Metric) sekä HVDM-funktio (Heterogeneous Value Difference Metric). HEOM-etäisyyden (10) arvo määräytyy lineaarisille muuttujille normalisoidun euklidisen etäisyyden mukaisesti. Nominaalisille muuttujille HEOM-etäisyys on joko 1 tai 0 sen mukaan, saavatko kaksi datapistettä saman vai eri arvon muuttujalle.

HEOM-funktio on muotoa

$$HEOM(x,y) = \sqrt{\sum_{i=1}^{a} d_i(x_i,y_i)^2} \text{ , missä} \quad (10)$$

$a = Muuttujien\ lukumäärä,$
$y = x{:}n\ naapuri\ ja$

$$d_i(x_i,y_i) = \begin{cases} 1, jos\ x_i\ tai\ y_i\ puuttuu \\ overlap(x_i,y_i) = \begin{cases} 0, jos\ x_i = y_i \\ 1, jos\ x_i \neq y_i \end{cases} \\ \dfrac{|x_i - y_i|}{\max_i - \min_i}, muulloin \end{cases}$$

(Wilson & Martinez, 1997)

HVDM-funktio (11) käyttää HEOM-funktiossa käytetyn overlap-suureen sijaan Stanfillin ja Waltzin (1986) esittelemää VDM-suuretta (Value Difference Metric), joka kuvaa muuttujien välistä erilaisuutta laskettuna muuttuja-arvojen frekvensseistä. HVDM-funktio on muotoa

$$HVDM(x,y) = \sqrt{\sum_{i=1}^{a} d_i(x_i,y_i)^2}\,, miss\ddot{a} \qquad (11)$$

$$d_i(x_i,y_i) = \begin{cases} 1, jos\ x_i\ tai\ y_i\ puuttuu \\ \sqrt{\sum_{q=1}^{c}\left|\frac{N_{i,x,q}}{N_{k,x}} - \frac{N_{i,y,q}}{N_{k,y}}\right|^2}\,, jos\ i\ on\ nominaalinen \\ \frac{|x_i - y_i|}{4\sigma_i}\,, muulloin \end{cases}$$

$N = Frekvenssi\ ja$
$\sigma_i = i{:}nnen\ muuttujan\ keskihajonta$

Kun datapisteitä luokitellaan lähimpien naapureiden luokkien perusteella, voidaan naapureille määrittää painoja siten, etteivät kaikki naapurit ole samanarvoisia luokittelun kannalta. Tällöin luokiteltavan datapisteen luokka äänestetään lähimpien naapureiden kesken siten, että kunkin naapurin ääni määräytyy etäisyyden lisäksi sen mukaan, kuuluuko naapuri kyseiseen luokkaan. Datapiste asetetaan eniten ääniä saaneeseen luokkaan. Käyttämällä äänestystä sen sijaan, että datapiste luokiteltaisiin suoraan naapuruston enemmistöluokkaan, saadaan vähennettyä kauimmaisten naapureiden vaikutusta luokitteluun verrattuna datapistettä lähimpinä oleviin naapureihin. Kullekin luokalle laskettava ääni voidaan määrittää etäisyyksien käänteislukujen summana (12), jossa p:tä kasvattamalla vähennetään kauempien naapureiden vaikutusta äänestykseen. (Dudani, 1976)

$$\ddot{a}\ddot{a}ni(c_j) = \sum_{i=1}^{k} \frac{1}{d(x,y_i)^p}\ 1(c_j, c_i,),\ miss\ddot{a} \qquad (12)$$

$$1(c_j, c_i,) = \begin{cases} 1, jos\ naapuri\ i\ kuuluu\ luokkaan\ j \\ 0, muulloin \end{cases}$$

Algoritmi 7: *k-NN(Z, X, k)*

Syöte: *Oikeat luokat sisältävä opetusaineisto Z, testiaineisto X, naapureiden lukumäärä k*

Tuloste: *K klusteria*

Jokaiselle testiaineiston alkiolle $x_i \in X$

1. *Laske etäisyys $d_{ij}(x_i,z_j)$ opetusaineiston jokaiseen alkioon $z_j \in Z$*
2. *Järjestä etäisyydet $d_{ij}(x_i,z_j)$ pienimmästä suurimpaan listaan l*
3. *Valitse k päällimmäisintä riviä järjestetystä listasta l*
4. *Valitse suurimman frekvenssin omaava luokka c_k*
5. *Palauta luokka c_k alkiolle x_i*

Pysähdy

MALLIN ARVIOINTI

K:n lähimmän naapurin onnistumista arvioidaan vertaamalla testiaineiston todellisia luokkia mallin ennustamiin luokkiin. Mallin tekemät ennusteet luokitellaan neljään luokkaan sen mukaan, onko ennuste todellinen poikkeavuus (true positive, TF), todellinen normaalihavainto (true negative, TN), normaalihavainto ennustettu poikkeavuudeksi (false positive, FT) vai poikkeavuus ennustettu normaaliksi havainnoksi (false negative, FT). Näiden luokkien frekvenssit esitetään nk. sekaannusmatriisina (confusion matrix), josta saadaan lasketua mallin tarkkuus (13), sensitiivisyys (14) sekä spesifisyys (15). Mallin tarkkuus kuvaa yleistä onnistumisprosenttia eli sitä, kuin suuren osan testiaineistosta malli onnistuu luokittelemaan oikein. Tarkkuus määritetään kaavan 13 mukaisesti

$$Tarkkuus = \frac{TP+TN}{TP+FP+FN+TN} \qquad (13)$$

Tarkkuuden lisäksi sekaannusmatriisista voidaan laskea sensitiivisyys, joka kuvaa mallin kykyä tunnistaa oikeat poikkeavuudet. Sensitiivisyys määritetään kaavan 14 mukaisesti

$$Sensitiivisyys = \frac{TP}{TP+FN} \qquad (14)$$

Mallin kykyä tunnistaa oikeat normaalit havainnot kuvaa spesifisyys, joka määritetään kaavan 15 mukaisesti

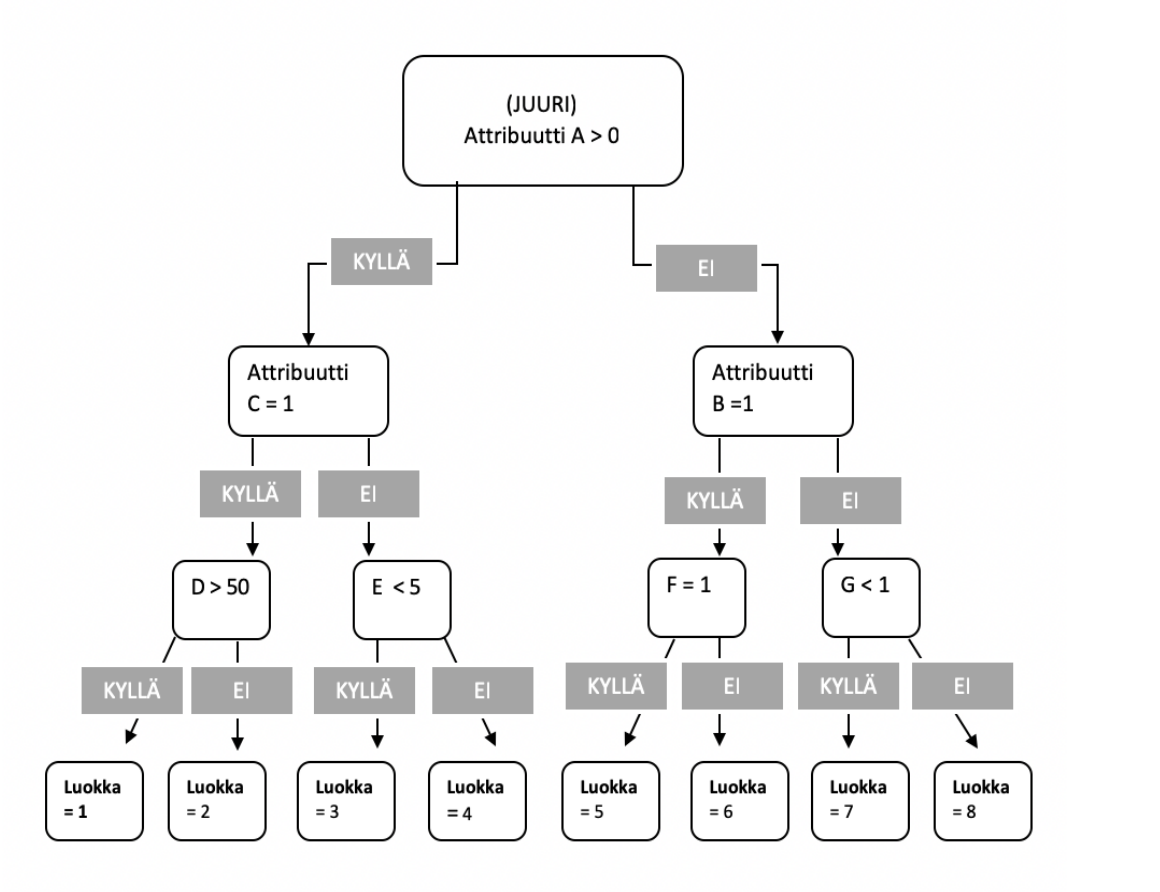
$$Spesifisyys = \frac{TN}{TN+FP} \qquad (15)$$

Mallin suorituskykyä arvioidessa voidaan painottaa yllä esitellyistä suureista jotakin tiettyä sen mukaan, minkälaisia virheitä luokittelijalta siedetään parhaiten. Mikäli halutaan varmistaa, että luokittelija osaa ennustaa mahdollisimman tarkasti kaikki todelliset poikkeavuudet, estimoidaan malli siten, että valikoidaan sellainen malli, joka tuottaa suurimman sensitiivisyyden. Tällöin todennäköisemmin myös normaaleja havaintoja ennustetaan poikkeavuuksiksi. Jos taas valitaan korkeimman spesifisyysarvon tuottava malli, vähenee ”väärien hälytysten” määrä, mutta toisaalta todellisia poikkeavuuksia voi jäädä tunnistamatta. (Cunningham ja Delany, 2007)

3.3.6 Päätöspuu

Päätöspuu (decision tree) on nimensä mukaisesti puurakenteena kuvattu sääntöpohjainen päättelyalgoritmi, jota koneoppimisessa käytetään sekä diskreettien muuttujien luokitteluun että jatkuvien muuttujien ennustamiseen regressiolla. Päätöspuita koskeva kirjallisuus keskittyy pääasiassa ohjatun oppimisen tehtäviin, mutta myös ohjaamattomissa tehtävissä päätöspuita on hyödynnetty (Liu, Xia ja Yu, 2005). Päätöspuu on

yleisimmin juurellinen binääripuu tarkoittaen, että jokaisella puun sisäsolmulla on tasan kaksi jälkeläistä ja oksat ovat yksisuuntaisia puun juuresta alas puun lehtiin, joilla ei ole yhtäkään jälkeläistä. Puun jokainen solmu sisältää päättelyn, joka ennustaa havainnon seuraavan solmun. Lopullinen ennuste, diskreetin luokittelijan tapauksessa havainnon luokka, sijaitsee puun lehdissä, joilla ei ole jälkeläisiä (kuva 4). Päätöspuun solmut voivat saada myös useampia kuin kaksi jälkeläistä, joten päätöspuu ei välttämättä ole binääripuu.



Kuva 4 Binäärinen päätöspuu etenee juuresta lehtiin

Päätöspuualgoritmi koostuu kahdesta vaiheesta; kasvattaminen ja karsiminen. Päätöspuu voidaan kasvattaa joko ”ylhäältä-alas” (top-down) tai ”alhaalta-ylös” (bottom-up) –menetelmällä. Algoritmin kasvatusvaihe alkaa juurisolmusta, joka sisältää kaikki datajoukon alkiot. Algoritmin ensimmäisessä vaiheessa datajoukko jaetaan kahteen osaan luokittelun kannalta tärkeimmän attribuutin suhteen valitun jakokriteerin perusteella. Syntyneet kaksi osajoukkoa sijoitetaan juuren jälkeläisiin, joista ne jaetaan edelleen osajoukkoihin seuraavien attribuuttien suhteen. Algoritmi etenee jakamalla osajoukkoja edelleen, kunnes joko pysähtymiskriteeri täyttyy tai data on jaettu jokaisen muuttujan suhteen. (Song & Lu, 2015)

Datajoukon jakaminen solmuista lehtiin tehdään valitun jakokriteerin mukaan. Jakokriteereistä suosituimpia ovat entropian muutoksen (16) tuottama informaatiolisä (Information gain) (17), Gini-kerroin (Gini Index)

(18) ja sen tuottama Gini-lisä (Gini gain) (19) sekä regressiopuun tapauksessa pienimmän neliövirheen summa (20). Entropia kuvaa datajoukon homogeenisuutta tai toisaalta varianssin suuruutta. Mitä pienempi datajoukon entropia-arvo on, sitä homogeenisempi datajoukko on kyseessä. Näin ollen entropian pienentyessä jaon jälkeen data on onnistuttu jakamaan kahteen osajoukkoon, joiden alkioiden samankaltaisuus on suurempi kuin alkuperäisen joukon. Entropia E solmussa N datajoukon S muuttujalle j määritellään ennustettavien luokkien todennäköisyyden käänteisluvun ja logaritmoitujen todennäköisyyksien tulon summana ja voidaan esittää muodossa

$$H(S_j) = -\sum_{i=1}^{c} p_i \log p_i\,,\ missä \qquad (16)$$

$c = Ennustettavien\ luokkien\ lukumäärä\ ja$
$p_i = Luokan\ i\ suhteellinen\ osuus\ muuttujassa\ j\ solmussa\ N$

Itse jakokriteeri on entropian muutoksen tuottama informaatiolisä, kun solmusta N tehdään jako muuttujan j suhteen. Mitä homogeenisemmaksi ennustettavan luokan suhteen jälkeläissolmut muuttuvat jaon jälkeen verrattuna solmuun N, sitä suurempi informaatiolisä saavutetaan. Informaatiolisä määritetään entropian muutoksena siten, että solmun N entropia-arvosta $H(S_j)$ vähennetään syntyvien jälkeläissolmujen entropia-arvojen painotettu keskiarvo kaavan 17 mukaisesti.

$$IG(S_j) = H(S_j) - \sum_{v}^{k} \frac{|S_{jv}|}{|S_j|} H(S_{jv}),\ missä \qquad (17)$$

$k = Jälkeläissolmujen\ v\ lukumäärä$

(Quinlan, 1993, s. 21–22)

Toinen päätöspuualgoritmeissa suosittu jakokriteeri on Gini-kertoimen tuottama informaatiolisä. Gini-kerroin kuvaa, kuinka sekoittuneita luokat ovat tehdyn jaon jälkeen. Pieni Gini-kerroin kuvastaa onnistunutta jakoa siten, että eri luokat ovat jakaantuneet selkeästi syntyneisiin osajoukkoihin. Gini-kerroin määritellään luokkien esiintymistodennäköisyyksien neliöiden summasta kaavalla

$$G(S_j) = 1 - \sum_{i=1}^{c} \left(\frac{|S_{ji}|}{|S_j|}\right)^2 \qquad (18)$$

Gini-kertoimen tuottama informaatiolisä kuvaa sitä, kuinka paljon hyötyä saavutetaan

$$GG(S_j) = G(S_j) - \sum_{i=1}^{c} G(S_j|i)\,, missä \qquad (19)$$

$\sum_{i=1}^{c} G(S_j|i) = Jaossa\ syntyneiden\ jälkeläissolmujen\ painotettujen\ Gini-kertoimien\ summa$

Yllä esitettyjen jakokriteereiden lisäksi regressiopuissa käytetään pienimmän neliövirheen summaa (20), joka kuvaa ennustevirheen suuruutta. Mitä pienempi neliövirheiden summa on, sitä paremmin arvot on onnistuttu ennustamaan, joten jakavaksi muuttujaksi valitaan pienimmän neliövirheen summan tuottava muuttuja. Neliövirheen summa voidaan esittää muodossa

$$SSE = \sum_{i=1}^{c} \sum_{x \in S_i} \left| |x - \bar{x}| \right|^2 \qquad (20)$$

Jakokriteerin lisäksi myös pysähtymiskriteerin määrittäminen on tärkeää, jotta päätöspuu ei kasvaisi liian suureksi. Mikäli päätöspuun syvyys eli juuresta lehtiin johtavan polun pituus, joka määritellään tehtyjen jakojen lukumääränä kasvaa liikaa, saattaa malli kärsiä ylisovituksesta. Tällöin malli luokittelee opetusaineiston tarkasti, mutta ei suoriudu hyvin uusien, ennen näkemättömien datapisteiden luokittelussa. Pysähtymiskriteereinä voidaan käyttää esimerkiksi puun maksimisyvyyttä, joka pysäyttää puun kasvatuksen, kun määrätty syvyys on saavutettu. Toinen mahdollinen pysähtymiskriteeri on solmuissa olevien alkioiden vähimmäismäärää, jolloin puun kasvatus pysähtyy, jos jaon seurauksena syntyneissä solmuissa alkioiden lukumäärä on tätä kynnysarvoa pienempi. (Drummond & Holte, 2000)

Pysähtymiskriteerin lisäksi päätöspuun ylisovitusta voidaan ehkäistä puun karsimisella. Puun karsinta voidaan suorittaa joko esikarsintana, jolloin luokittelun kannalta epäolennaisimmat oksat karsitaan jo kasvatusvaiheen yhteydessä, tai jälkikarsintana vasta puun kasvatuksen jälkeen. Jälkikarsinnassa sellaiset kasvatetun päätöspuun alipuut, joiden hyöty luokittelun tarkkuuden kannalta on vähäisin, korvataan lehtisolmuilla. (Song ja Lu, 2015)

Vertailussaan eri karsintamenetelmistä Mingers (1989) esittää hyviksi menetelmiksi kriittisen arvon, virhekompleksisuus sekä vähentyneen virheen menetelmiä. Kriittisen arvon esikarsintamenetelmässä karsitaan solmut, jotka eivät saavuta valittua solmun tärkeyttä mittaavaa kriittistä arvoa. Tällöin kasvatettu puu jää sitä pienemmäksi, mitä suurempi kriittinen arvo valitaan.

Virhe-kompleksisuuskarsinta on kaksivaiheinen menetelmä, jossa jälkikarsinta tehdään sekä virheiden lukumäärän että puun koon perusteella. Karsinta aloitetaan luomalla joukko karsittuja puita poistamalla alipuita siten, että korvattaessa alipuu lehdellä lehden luokaksi määräytyy alipuun lehtien yhteenlaskettu enemmistöluokka. Virheaste määräytyy niiden havaintopisteiden lukumäärästä, jotka eivät kuulu tähän enemmistöluokkaan. Menetelmän ensimmäinen vaihe päättyy virhe-kompleksisuusarvon laskentaan, joka määritellään jakamalla karsitun puun virheasteen erotus alkuperäisen puun virheasteeseen puun lehtien lukumäärällä. Toisessa vaiheessa valitaan pienimmän virheen tuottava alipuu ensimmäisessä vaiheessa muodostettujen puiden joukosta. Virhetermi muodostetaan testiaineiston luokittelemisella, sillä opetusaineiston luokittelulla

valittaisiin aina alkuperäinen karsimaton puu, joka alun perin kasvatettiin samaisella opetusaineistolla. (Breiman, Friedman, Olshen ja Stone, 1984)

Vähentyneen virheen menetelmä on jälkikarsinta, jossa opetusaineistolla kasvatettu puu karsitaan testiaineistoa käyttäen. Menetelmässä puun alipuu korvataan lehtisolmulla silloin, jos testidatan luokittelussa virheellisten luokitusten määrä vähenee korvaamalla alipuu lehtisolmulla. (Mingers, 1989) Edellä esiteltyjen karsintamenetelmien lisäksi nykyisin paljon käytetty esikarsinta menetelmä, chi-toiseen-karsinta, perustuu nimensä mukaan χ^2-testiin. Karsinnassa mitataan, onko jaosta syntyvien solmujen homogeenisuus merkittävästi lähtösolmua suurempi. Mikäli ero ei ole tilastollisesti merkittävä, ei jakoa tehdä. (Patel ja Upadhyay, 2012)

Päätöspuualgoritmeja on kehitetty lukuisia erilaisia eri informaatiokriteereille. Käytetyimpiä päätöspuualgoritmeja lienevät ID3, C4.5 sekä CART. ID3-algoritmi, jonka esitteli Ross Quinlan vuonna 1979 (Quinlan, 1986), kuuluu niin kutsuttuun TDITD-perheeseen (Top-Down Induction of Decision Tree), jonka algoritmeille on ominaista kasvattaa puu juuresta kohti lehtiä. ID3 on iteratiivinen menetelmä, jossa perusajatuksena on kasvattaa opetusaineiston osajoukkoa käyttäen sellainen puu, joka luokittelee oikein koko opetusaineiston. Opetusaineistosta poimitaan ”ikkuna”-osajoukko, ja puu kasvatetaan siten, että kukin ikkunajoukon havainnoista luokitellaan oikeaan luokkaan. Tämän jälkeen ikkunan ulkopuoliset opetusaineiston havainnot luokitellaan kasvatetulla puulla. Ikkunan ulkopuoliset havainnot, jotka puu luokittelee virheellisesti, lisätään ikkunajoukkoon ja kasvatetaan puu uudelleen. Tämä toistetaan, kunnes on kasvatettu puu, joka luokittelee oikein koko opetusaineiston. Algoritmi 7 kuvaa Quinlanin (1986) esittelemän ID3-algoritmin toimintaperiaatteen. C4.5 sekä C5.0 ovat ID3-algoritmin myöhempiä muunnelmia, joiden avulla on pyritty ylittämään ID3:n suurimpia haittapuolia. Toisin kuin ID3, C4.5 pystyy käsittelemään myös jatkuvia muuttujia. Koska ID3 kasvattaa päätöspuun niin syväksi, että jokainen opetusaineiston havainto luokitellaan oikein, on ylisovitus sen suurimpia ongelmia. C4.5-algoritmi kattaa puun kasvatuksen lisäksi karsintavaiheen, jonka avulla ylisovitus ei ole yhtä ilmeinen ongelma. C4.5 pystyy myös käsittelemään jatkuvia muuttujia ja puuttuvia arvoja sekä painottamaan eri muuttujia. C4.5-algoritmin ennustetarkkuutta, luokittelun tulkittavuutta sekä nopeutta on paranneltu versioon C5.0. (Hssina, Merbouha, Ezzikouri ja Erritali, 2014)

CART-algoritmi on Leo Breimanin (1984) kehittämä päätöspuupohjainen algoritmi, jota voidaan käyttää luokittelun lisäksi myös jatkuvan kohdemuuttujan ennustamiseen, regressioon. Erona ID3-algoritmiin ja sen muunnelmiin on, että CART kasvattaa binäärisen päätöspuun ja jakokriteerinä on entropian muutoksen sijaan Gini-kertoimen muutoksen tuottama Gini-lisä. (Hssina et al., 2014)

Algoritmi 8: *ID3(C, A)*

Syöte: *Oikeat luokat sisältävä opetusaineisto C, attribuutit A*

Tuloste: *Päätöspuu, joka luokittelee jokaisen opetusaineiston havainnon oikeaan luokkaan*

1. *Luo puun juurisolmu*
2. *a. Jos kukin juuren alkioista kuuluu luokkaan c*
   Palauta lehtisolmu luokalla c
   b. Muutoin, jos attribuuttijoukko A on tyhjä
   Palauta lehtisolmu
   c. Muutoin, jatka vaiheeseen 3
3. *Valitse sellainen attribuutti $a_i \in A$, joka maksimoi entropian muutoksen informaatiolisän*
   a. *Jokaiselle a_i:n arvolle v_j*
   Lisää oksa $a_i = v_j$ ja aseta osajoukko $C_{ai = vj} \in C$ solmuun n
   b. Jos $C_{ai = vj} = \emptyset$
   Palauta lehtisolmu
   Muutoin
   Toista vaihe 3, kunnes koko C on luokiteltu oikeisiin luokkiin

Pysähdy

Algoritmi 9: *CART(C, A, x)*

Syöte: *Oikeat luokat sisältävä opetusaineisto C, muuttujat A, pysähtymiskriteeri x*

Tuloste: *Binääripäätöspuu*

1. *Jokaiselle muuttujalle $a_i \in A$*
   Etsi arvo a_{imax}, joka maksimoi jakokriteerin
   Järjestä A listaan l laskevaan järjestykseen arvojen a_{imax} mukaan
2. *Jaa solmu listan l ensimmäisen alkion arvon a_{imax} mukaan*
3. *Kunnes pysähtymiskriteeri x kohdataan*
   Toista kohtia 1. ja 2.
4. *Suorita karsinta*

Pysähdy

3.3.7 Satunnaismetsä

Yksittäisillä päätöspuilla on todettu olevan joitakin rajoituksia ja heikkouksia luokittelijoina. Yksittäinen päätöspuu kasvattaa aina parhaan mahdollisen puun käytettävissä olevan opetusaineiston perusteella siten, että jokaisen solmun kohdalla jako tehdään sellaisen muuttujan suhteen, joka parhaiten pystyy jakamaan jäljellä olevan datan kahteen luokkaan. Mitä syvemmäksi päätöspuu kasvaa sitä tarkemmin se kykenee luokittelemaan opetusaineiston havainnot ja sen myötä ylisovittamisen riski kasvaa. Seurauksena kasvatettu päätöspuu saattaa onnistua erinomaisesti opetusaineiston luokittelussa, mutta testiaineiston luokittelussa huomattavasti heikommin.

Päätöspuu on herkkä sille, minkä muuttujan suhteen ja minkä arvon perusteella kukin jako tehdään sillä kukin jako vaikuttaa syvemmällä puussa tehtäviin jakoihin. Tätä ominaisuutta kuvaa päätöspuun tendenssi suureen varianssiin, jonka seurauksena algoritmi on herkkä muutoksille aineistossa. Jos esimerkiksi jonkun luokittelun kannalta tärkeän selittävän muuttujan arvot muuttuvat, saattaa myös puun rakenne muuttua merkittävästi.

Päätöspuun ongelmia on ratkaistu useammilla algoritmimuunnoksilla, joista yksi suosituimmista on satunnaismetsä (Random Forest). Satunnaismetsä on päätöspuupohjainen algoritmi, jossa yksittäisen puun sijaan kasvatetaan useampia puita, joista päätellään ennustettu luokka enemmistöäänestyksellä. Satunnaismetsä ratkaisee päätöspuun ongelmat, mitä tulee aineistosensitiivisyyteen. Yksittäinen päätöspuu voi olla hyvinkin herkkä muutoksille opetusaineistossa, jolloin opetusaineiston muuttuessa kasvatettu puu voi muuttua paljonkin. Satunnaismetsän kohdalla kasvatettavia puita on useita, jolloin tällaista ongelmaa ei pääse niin herkästi muodostumaan. Lisäksi satunnaismetsän kohdalla ylisovituksen riski ei ole yhtä suuri kuin yksittäisillä päätöspuilla, kun luokka ennustetaan koko metsän puiden ennusteet huomioiden. (Breiman, 2001)
Breimanin algoritmin satunnaisuus luodaan kasvattamalla useita päätöspuita, joiden jokaisen solmun jakava muuttuja valitaan satunnaisesti muodostetusta muuttujien osajoukosta. Näin ollen osajoukko, josta jaon tekevä muuttuja valitaan, on satunnainen ja tästä joukosta valitaan luokittelun kannalta paras muuttuja valitun jakokriteerin perusteella. Muuttujajoukon satunnainen valinta vähentää aggregoitavien puiden välistä korrelaatiota ja näin tuloksena saadaan satunnaisia ja toisistaan riippumattomia puita eli satunnaismetsä. Kun metsä on kasvatettu, valitaan ennustettu luokka enemmistöäänestyksellä.

Menetelmäksi satunnaismetsän satunnaisuuden muodostamiseen on esitetty useita vaihtoehtoja, kuten satunnaisjakovalinta (random split selection) (Dieterrich, 1999), Adaboost (Freund ja Schaphire, 1996) sekä nykyisin käytetyin boosting-aggregation eli bagging (Breiman, 2001). Breiman (2001) osoitti työssään, että yhdistämällä bagging sekä syötemuuttujien satunnainen valinta (random input selection) satunnaismetsä ensinnäkin tuottaa pienimmän testivirheen (test set error), mutta lisäksi metsät ovat huomattavasti nopeampia kasvattaa kuin esimerkiksi Adaboost-menetelmää käytettäessä.

Satunnaismetsäalgoritmi alkaa yksittäisten päätöspuiden kasvattamisella. Jokaista päätöspuuta kohden valitaan satunnainen otos alkioita opetusdatasta bootstrap-menetelmällä. Kullekin puulle valittu opetusaineisto C_j^* on yhtä suuri kuin alkuperäinen opetusaineisto C. Jos opetusaineistossa C on n alkiota, valitaan kunkin päätöspuun opetusaineistoon C_j^* n alkiota bootstrap-menetelmällä tarkoittaen, että mielivaltaisesti valittu alkio palautetaan takaisin alkuperäiseen datajoukkoon siten, että sillä on yhtä suuri todennäköisyys tulla valituksi uudestaan osajoukkoon C_j^* kuin kaikilla muilla opetusaineiston C alkioilla. Näin ollen samat alkiot voivat esiintyä puun j opetusaineistossa C_j^* useaan kertaan, vaikka alkuperäisessä opetusaineistossa C ne esiintyisivät vain kertaalleen. (Breiman, 2001)

Puukohtaisen opetusaineiston C_j^* poiminnan jälkeen päätöspuuta lähdetään kasvattamaan juuresta lehtiin siten, että kunkin solmun kohdalla valitaan satunnaisesti valitusta muuttujien osajoukosta muuttuja, joka on jakokriteerin perusteella paras jakava muuttuja. Satunnaisesti valittu muuttujien osajoukko valitaan kaikkien muuttujien joukosta palauttamatta siten, että kukin muuttuja voi esiintyä joukossa vain kertaalleen. Muuttujien lukumäärä on aina pienempi kuin koko muuttuja-avaruus. Kun päätöspuita on kasvatettu haluttu määrä, asetetaan luokiteltavan havainnon luokaksi kasvatetun metsän enemmistöluokka. (Breiman, 2001)

Algoritmi 9: *Satunnaismetsä(C, X, k)*

Syöte: *Oikeat luokat sisältävä opetusaineisto C, testiaineisto X, puiden lukumäärä k*

Tuloste: *k:n puun satunnaismetsä*

1. *Muodosta k boostrap-otosta C_k^* opetusaineistosta C*
2. *a. Valitse muuttujajoukosta satunnaisesti m muuttujaa*
   b. Jokaiselle otokselle C_k^* kasvata karsimaton päätöspuu valitsemalla paras jakaja joukosta m
3. *Palauta instanssin luokka k:n puun enemmistööänestyksen perusteella*

Pysähdy

Satunnaismetsän suoriutumista arvioidaan OOB(Out-of-Bag)-virhetermin, vahvuuden sekä korrelaation perusteella ja tavoitteena on minimoida metsän tuottama virhe. Bernard, Heutte ja Adam (2010) osoittavat työssään, että virheen pienetessä satunnaismetsän vahvuus kasvaa ja yksittäisten puiden välinen korrelaatio pienenee. OOB-virhetermi määritetään bootstrap-otoksen ulkopuolelle jääneitä otoksia käyttäen siten, että kunkin opetusaineiston otoksen C_k^* keskimääräinen virhetermi lasketaan vain sellaisia puita käyttäen, joiden kasvatuksessa ei käytetty otosta C_k^*. (Breiman, 1996) Breiman (2001) johtaa yksittäisten puiden vahvuuden sekä puiden välisten korrelaatioiden keskiarvon avulla ylärajan metsän yleistysvirheelle (21), joka suppenee kohti nollaa puiden lukumäärän kasvaessa. Puun vahvuus kuvaa sen marginaalifunktion odotusarvoa. Metsän yleistysvirheen yläraja on muotoa

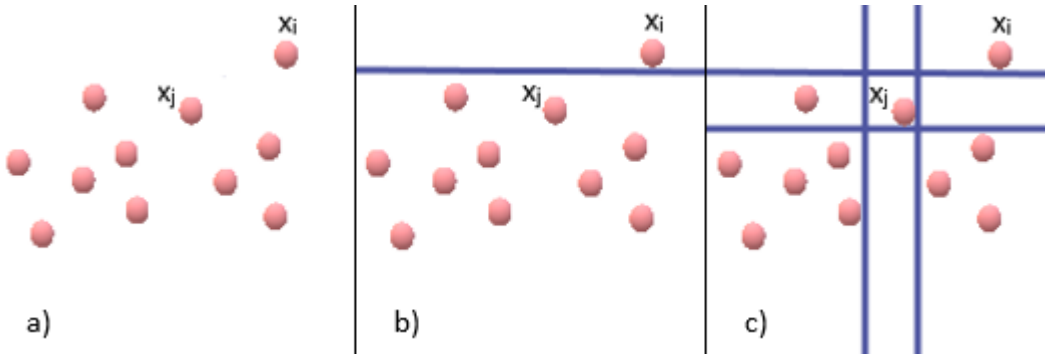
$$PE^* \leq \frac{\bar{\rho}(1-s^2)}{s^2}, missä \qquad (21)$$

$\bar{\rho} = Keskimääräinen\ korrelaatio\ ja\ s = Vahvuus$

Epäyhtälöstä voidaan edelleen johtaa metsän suorituskyvyn arvioimiseen käytettävä keskimääräisen korrelaation ja vahvuuden neliön suhde $\frac{\bar{\rho}}{s^2}$, joka saa sitä pienemmän arvon, mitä paremmin metsä suoriutuu. (Breiman, 2001)

3.3.8 Eristysmetsä

Eristysmetsä (isolation forest, iForest) on satunnaismetsän tapaan päätöspuupohjainen algoritmi, jonka Liu, Ting sekä Zhou (2009) kehittivät nimenomaan poikkeavien havaintojen tunnistamiseen. Kolmikko osoitti työssään, että eristysmetsä suoriutuu poikkeavuuksien tunnistuksesta etäisyys- ja tiheysperusteisia menetelmiä paremmin etenkin suurten ja monidimensioisten aineistojen kohdalla. Eristysmetsän vahvuutena verrattuna etäisyys- ja tiheysmittoihin perustuviin menetelmiin on tunnistaa myös ryhmittäisiä poikkeavuuksia pistemäisten poikkeavien havaintojen lisäksi. Etäisyys- ja tiheysmittoihin perustuvat menetelmät havainnoivat poikkeavuuksiksi datapisteet, jotka ovat joko etäällä muusta datajoukosta tai joiden läheisyydessä on vähän muita datapisteitä. Tällaiset menetelmät eivät kykene tehokkaasti tunnistamaan poikkeavuuksien ryhmittymiä, joissa datapisteet sijaitsevat lähekkäin tai tiiviisti toisiinsa nähden. Eristysmetsä käyttää etäisyys- ja tiheyssuureiden sijaan poikkeavuuksien määrittämiseen polun pituutta. Eristysmetsä erottelee (eristää) havainnot toisistaan tekemällä jakoja kuten päätöspuu- tai satunnaismetsäalgoritmitkin. Se, kuinka monta jakoa on tehtävä ennen kuin alkio on eristetty muusta datajoukosta, kuvaa alkiokohtaista polun pituutta. Mitä lyhyempi polku on, sitä todennäköisemmin datapiste on poikkeavuus olettaen, että poikkeavuudet on helpompi erottaa muusta datajoukosta kuin normaalit datapisteet (kuva 5).



Kuva 5 b) Datapisteen x_i eristäminen vaati vain yhden jaon, kun taas c) pisteen x_j eristäminen vaatii neljä jakoa

Eristysmetsä koostuu joukosta binäärisiä eristyspuita (iTree). Opetusvaiheessa eristyspuut kasvatetaan käyttäen opetusaineiston osajoukkoja. Puuta kasvatettaessa valitaan kunkin solmun kohdalla muuttujajoukosta satunnainen muuttuja, jonka suhteen jako tehdään satunnaisen jakoarvon perusteella. Puun kasvatusta jat-

ketaan, kunnes kaikki datajoukon havainnot on eristetty toisistaan. Opetusvaiheessa kasvatettua metsää käytetään prosessin seuraavan vaiheen, arvioinnin, syötteenä. Arviointivaiheessa lasketaan metsän keskimääräinen polun pituus kullekin testiaineiston datapisteelle. Laskettua polun pituutta käytetään edelleen datapisteiden poikkeavuusarvojen määrittämiseen. Poikkeavuusarvon perusteella alkiot lopulta luokitellaan joko poikkeavuuksiksi tai normaaleiksi. Polun pituuden h(x) yksikkö on tehtyjen jakojen lukumäärä, joka määritetään metsälle yksittäisten puiden keskiarvona E(h(x)). Poikkeavuusarvon s määrityksessä pituus E(h(x)) standardoidaan epäonnistuneiden hakujen keskiarvolla kaavan 22 mukaisesti

$$s(x, N) = 2^{-\frac{E(h(x))}{c(N)}}, missä \qquad (22)$$

$s(x, N) = Datapisteen\ x\ poikkeavuusarvo,$
$E(h(x)) = polun\ pituus,$
$c(N) = Epäonnistuneiden\ hakujen\ polun\ keskipituus\ ja$
$N = Osajoukon\ datapisteiden\ lukumäärä$

(Liu, Ting ja Zhou, 2009)

Algoritmi 10: *Eristysmetsä: Opetus (C, k, N)*

Syöte: *Opetusaineisto C, puiden lukumäärä k, osajoukon koko N*

Tuloste: *K:n puun eristysmetsä*

1. *Jokaiselle j:n arvolle 1-k:*
   Poimi joukosta C N:n alkion osajoukko $C_j´$
2. *Jokaisesta osajoukosta Cj´ kasvata eristyspuu Puu_j:*
   a. *Valitse satunnainen attribuutti q∈Q, Q = Attribuuttien joukko*
   b. *Valitse satunnainen jakoarvo p attribuutin q arvojoukosta*
   c. *Jaa $C_j´$ vasempaan ja oikeaan jälkeläiseen q:n ja p:n perusteella*
3. *Lisää Puu_j eristysmetsään*

Pysähdy

Algoritmi 11: *Eristysmetsä: Arviointi(x, K, h_{max}, e)*

Syöte: *Alkio x, eristyspuu K, maksimikorkeus h_{max}, nykyinen polun pituus e*

Tuloste: *Alkion x polun pituus*

1. *Jos e ≥ h_{max} tai ollaan T:n lehtisolmussa:*
   a. *Pysähdy ja palauta e + c(N)*
2. *Muutoin aseta*
   a. *q ← jakoattribuutti*
3. *Jos x_q< p, p = jakoarvo*
   a. *Palauta pituus(x, K_{vasen}, h_{max}, e+1)*
4. *Muutoin*

a. *Palauta pituus(x, K_{oikea}, h_{max}, e+1)*

Pysähdy

4. TUTKIMUSAINEISTO

Tutkimusaineisto on monidimensionaalinen sekatyyppinen aineisto, joka sisältää myös puuttuvia havaintoja. Jotta koneoppimismallit pystyisivät tunnistamaan poikkeavat havainnot mahdollisimman tarkasti, tulee aineiston esikäsittely harkita huolellisesti. Tässä luvussa kuvataan tutkimusaineiston rakenne sekä aineistolle suoritetut esikäsittelyt.

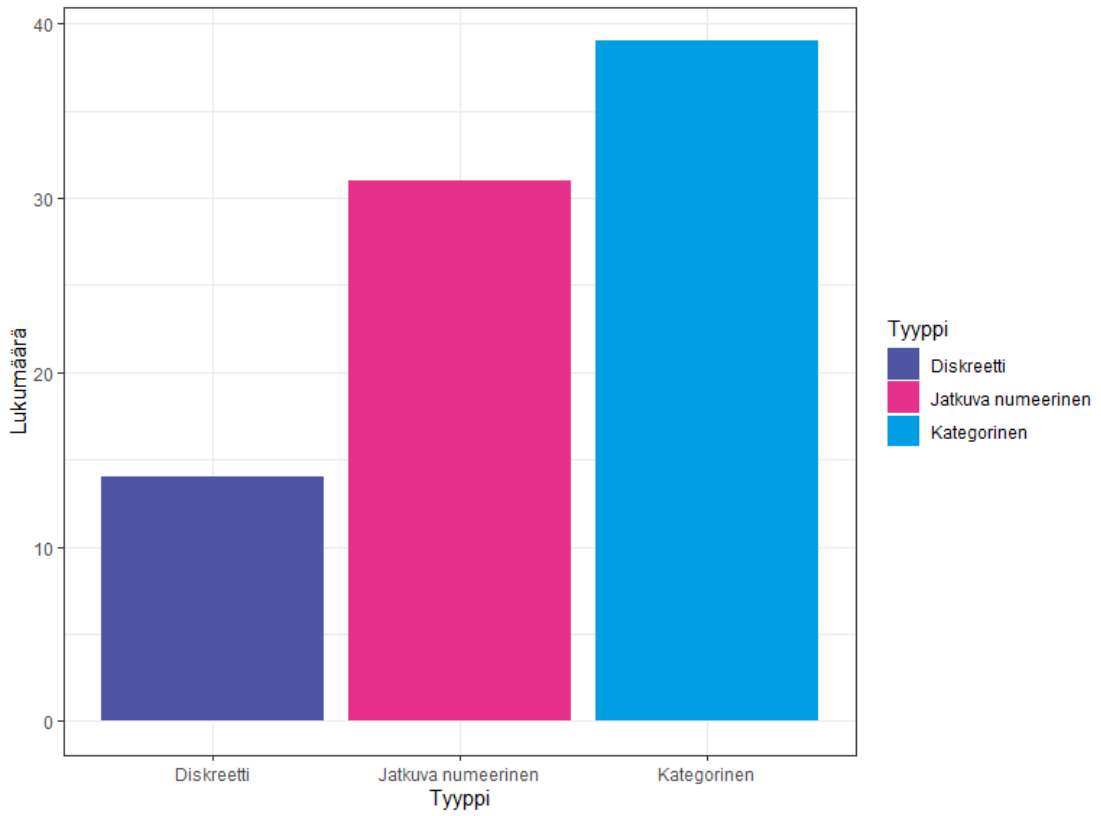
4.1 Aineiston kuvaus

Tutkimusaineisto koostuu Suomen Pankin tuottamasta rahalaitosten tase- ja korkotilastosta. Tilasto kerätään kuukausittain suurimmilta Suomessa toimivilta rahalaitoksilta sekä vuosineljänneksittäin pienemmiltä toimijoilta. Kvartaalikuukausien väliin jääville kuukausille neljännesvuosiraportoijien raportit kopioidaan, joten raporttilukumäärä ei muutu neljännes- ja välikuukausien välillä. Tase- ja korkotilasto sisältää tiedot rahalaitosten taseen vastaavista ja vastattavista aggregaattitasolla. Näin ollen tiedonkeruussa ei raportoida esimerkiksi laina- tai talletustilikohtaisia tietoja vaan kaikki toisiaan vastaavat tase-erät aggregoidaan yhdeksi datapisteeksi.
Tutkimusaineiston ulkopuolelle rajataan sekä arvopaperit, johdannaiset että taseen ulkopuoliset erät. Lisäksi erillisellä tietueella raportoitavat arvon alentumiset jätetään pois tarkastelusta ja aineisto rajataan sisältämään taseen saamispuoli. Lopullinen tutkimusaineisto on tase- ja korkotilaston alaerä sisältäen tiedot rahalaitosten taseessa olevista saamisista.

Aineisto koostuu noin 250:sta raportista, joista kukin sisältää rivejä (datapisteitä) pienimpien raportoijien muutamasta rivistä suurimpien raportoijien kymmeniin tuhansiin riveihin. Kokonaisrivimäärä vaihtelee hieman kuukausittain ollen puolen miljoonan molemmin puolin keskimääräisenä kuukautena. Käsittelemätön aineisto sisältää yhteensä 91 dimensiota, joista osa on raportoituja tietoja ja osa saa arvonsa Suomen Pankin rikastuksista. Suomen Pankki muun muassa rikastaa vastapuolitietoja rekistereistä sekä laskee raporteille numeerisia kenttiä, kuten tase-erien nettomääräiset virtatiedot. Lisäksi aineistoon on tuotu raporttikohtaisia lisätietoja raportoijan ja raportin tiedoista, kuten raportin versiotiedot. Mallien sovitukseen käytetään joulukuun 2017 aineistoa.

Aineisto sisältää sekä kvalitatiivisia että kvantitatiivisia muuttujia, joten kyseessä on niin kutsuttu sekatyyppinen aineisto. Numeeriset muuttujat ovat pääasiassa jatkuvia, kuten tase-erän nettomääräinen virta, koron arvo tai tasearvo. Mukana on kuitenkin myös muutamia diskreettejä muuttujia, kuten koron kiinnitysaika ja lainan kokoluokka. Alkuperäisessä aineistossa on mukana joitakin ylimääräisiä muuttujia, jotka ovat tarpeel-

lisia vain dataa haettaessa sql-tietokannan eri tietokantatauluista, mutta eivät tutkimusongelman näkökulmasta ole mielenkiintoisia. Tällaiset muuttujat poistetaan heti haun suorittamisen jälkeen. Lisäksi aineistossa on sellaisia muuttujia, jotka saavat arvoja ainoastaan taseen velkapuolella, joten myös tällaiset muuttujat jäävät tarkastelun ulkopuolelle. Muuttujapoistojen jälkeen aineistoon jäi 84 muuttujaa, joista 31 jatkuvaa, 14 diskreettiä ja 39 kategorista (kuva 6).



Kuva 6 Tutkimusaineiston muuttujien lukumäärät tyypeittäin

4.2 Esikäsittely

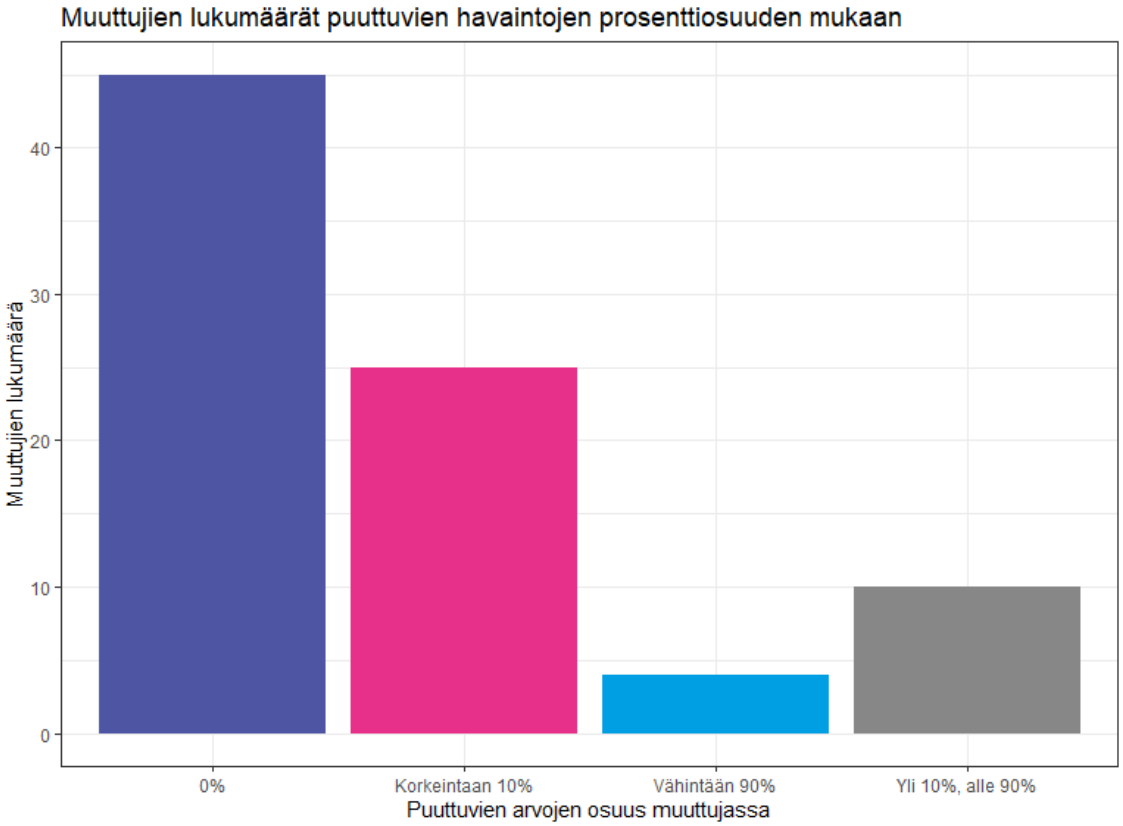
Esikäsittely on kriittinen vaihe koneoppimismallien rakentamisessa, ja sen huolellinen toteuttaminen varmistaa menetelmien optimaalisen suoriutumisen. Esikäsittelemätön aineisto saattaa johtaa paitsi harhaisiin tuloksiin, voi joidenkin menetelmien kohdalla olla jopa mahdotonta ajaa algoritmia ilman muuttujien esikäsittelyä. Etenkin kategoristen muuttujien ja puuttuvien arvojen käsittely tulee tehdä harkiten, jotta algoritmin suoriutuminen saadaan taattua.

4.2.1 Kategoristen muuttujien käsittely

Suurin osa koneoppimismenetelmistä ei pysty käsittelemään kategorisia muuttujia sellaisenaan, vaan muuttujat pitää muuntaa algoritmille ymmärrettävään muotoon. Tutkimusaineiston kategoriset muuttujat käsitellään ”one-hot”-koodauksella (one-hot encoding) siten, että kunkin muuttujan kullekin arvolle muodostetaan uusi binäärinen muuttuja. Tällöin kyseinen uusi muuttuja saa arvon 1 silloin, kun alkuperäinen muuttuja saa tämän arvon ja arvon 0 muulloin. Kategoristen muuttujien käsittelyn jälkeen kukin käsitelty alkuperäinen muuttuja poistetaan. Kategoristen muuttujien esikäsittelyn jälkeen aineistossa on 187 muuttujaa.

4.2.2 Puuttuvien havaintojen käsittely

Aineisto sisältää sekä muuttujia, joiden ei ole mahdollista sisältää puuttuvia arvoja että muuttujia, joilta pääsääntöisesti arvo puuttuu. Puuttuvien arvojen käsittely jaetaan kolmeen osaan sen mukaan, kuinka tyypillisiä puuttuvat arvot muuttujille ovat. Muuttujia, jotka ovat pakollisia raportoida eivätkä näin ollen vaadi puuttuvien arvojen käsittelyä, on tutkimusaineistossa 45. Sellaisia muuttujia, joilla puuttuvat arvot ovat harvinaisia, on 25. Näiden lisäksi aineistossa on muutama muuttuja, jotka saavat arvon vain harvoin (puuttuvia arvoja yli 90 %) ja 10 muuttujaa, joilla puuttuvia arvoja esiintyy jonkin verran, mutta eivät kuitenkaan pääsääntöisesti ole puuttuvia (kuva 7).



Kuva 7 Muuttujista noin kolmasosa sisältää puuttuvia havaintoja

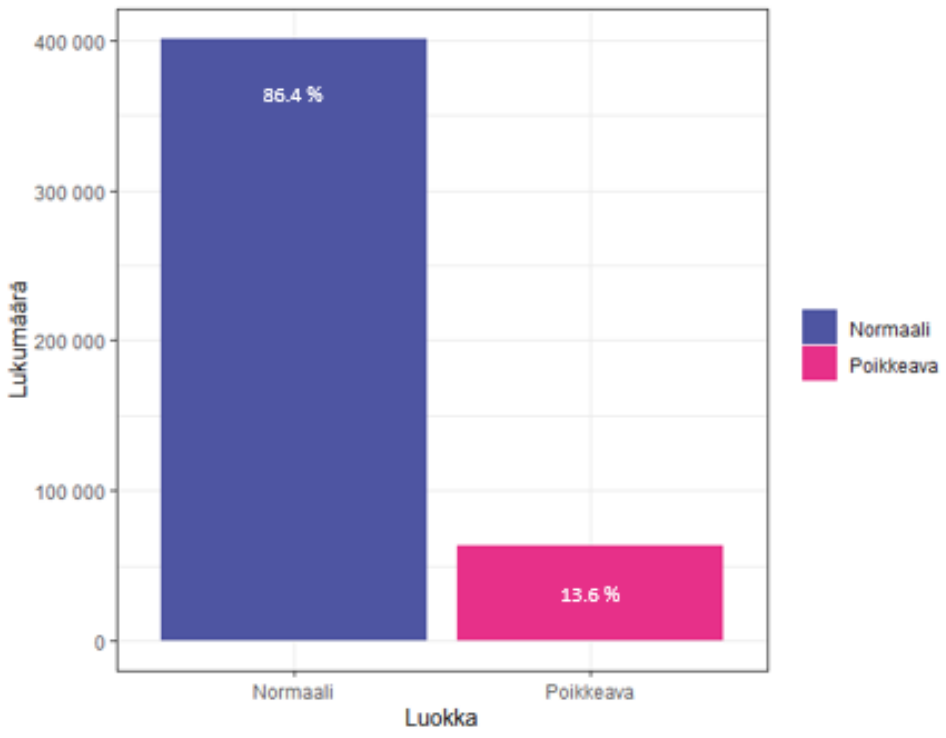
Niiden muuttujien kohdalla, joiden arvot ovat pääsääntöisesti puuttuvia (puuttuvien arvojen osuus yli 90 %), poistetaan alkuperäinen muuttuja ja lisätään tilalle binäärinen muuttuja, joka saa arvokseen 1 mikäli arvo on puuttuva ja 0 mikäli muuttuja saa arvon.

Alle kymmenen prosenttia puuttuvia arvoja sisältävistä muuttujista suurin osa on sellaisia, jotka kuvaavat jonkun toisen muuttujan arvoa edellisellä periodilla. Datapisteet, jotka ovat uusia aineistossa, eivät näin ollen saa arvoa näille muuttujille. Tällaisten muuttujien puuttuvat arvot käsitellään asettamalla muuttujalle arvo, jonka vastaava tarkasteluperiodin muuttuja saa. Tällöin tarkasteluperiodin ja edellisen periodin arvojen erotukseksi saadaan nolla, joka vastaa parhaiten tilannetta. Muut alle kymmenen prosenttia puuttuvia arvoja sisältävät muuttujat käsitellään samaan tapaan kuin yli 90 % puuttuvia sisältävät muuttujat. Näin tehdään siitä syystä, että merkittäväksi tiedoksi ajatellaan nimenomaan se, että arvo on puuttuva.

Loput puuttuvia havaintoja sisältävät muuttujat käsitellään tapauskohtaisesti sen mukaan, minkä tyyppinen muuttuja on kyseessä. Mikäli muuttuja saa pääsääntöisesti aina arvokseen nolla, mutta joukossa on muutama nollasta merkittävästi poikkeava arvo, asetetaan puuttuvien arvojen tilalle nolla. Muussa tapauksessa numeeristen muuttujien puuttuvat havainnot korvataan joko muuttujan keskiarvolla tai mediaanilla.

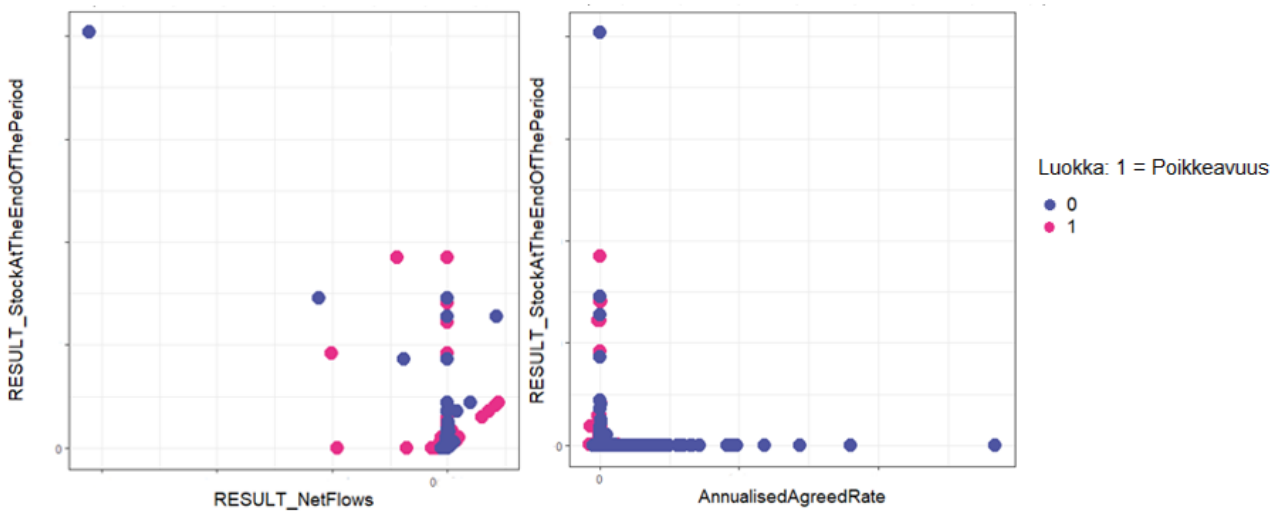
4.2.3 Poikkeavat havainnot

Jotta poikkeavia havaintoja voidaan etsiä aineistosta ohjatun oppimisen menetelmin, täytyy opetusaineistolle kerätä poikkeavuusluokitus. Suomen Pankin tietokannasta on haettavissa kaikki saapuneet raporttiversiot ja näiden versiotietojen avulla poikkeavat havainnot saadaan poimittua aineistoon. Poikkeavien havaintojen tunnistus tutkimusaineistosta suoritetaan vertaamalla raportoijittain ensimmäistä saapunutta raporttiversiota viimeiseen (lopulliseen) versioon. Aineistoon muodostetaan uusi muuttuja (”Outlier”) poikkeavuusluokalle siten, että muuttuja saa arvon 1 mikäli datapiste (rikastetun raportin rivi) on muuttunut ensimmäisen ja viimeisen raporttiversion välillä ja arvon 0, mikäli datapiste on säilynyt muuttumattomana. Lopullinen aineisto koostetaan ensimmäisten raporttiversioiden havainnoista lisättynä viimeisen raporttiversion avulla muodostettu poikkeavuusluokka. Aineistoa kootessa olisi mahdollista valita mukaan ainoastaan ne raportit, joilta on saapunut useampi kuin yksi versio. Näin saataisiin nostettua poikkeavien havaintojen prosentuaalista osuutta koko aineistosta, mikäli poikkeavuuksien määrä ei muuten riittäisi siihen, että menetelmät oppisivat tunnistamaan poikkeavuudet normaaleista havainnoista. Tässä tapauksessa poikkeavuuksien määrä kasvaisi kuitenkin odottamattoman suureksi (yli 50 %). Toisaalta poikkeavuuksien osuus koko aineistosta on yli 10 % (kuva 8), jonka pitäisi pääsääntöisesti riittää oppimiseen, joten raporttijoukko päätetään jättää karsimatta.



Kuva 8 Poikkeavien havaintojen osuus aineistossa on 14 %

Tärkeimpien jatkuvien muuttujien suhteen tarkasteltuna (kuva 9) huomataan, että markkina-arvon (RESULT_StockAtTheEndOfThePeriod) ja virran (RESULT_NetFlows) suhteen poikkeavat ja normaalit luokat jakautuvat melko tasaisesti ääriarvoille, mutta poikkeavuuksia on kuitenkin selvästi erotettavissa ääriarvoista. Kaikkein äärimmäisin arvo kuuluu normaaliin luokkaan. Havainto kuvastaa, kuinka tilastollisessa aineistossa ääriarvot eivät itsessään aina ole virheellisiä. Tämän ominaisuuden vuoksi tilastollisen aineiston laadunvalvontaprosessissa ei riitä, että ainoastaan ääriarvot tarkistettaisiin, koska suuri osa poikkeavuuksista jakaantuu melko tasaisesti jatkuvien muuttujien eri arvoille. Näin ollen poikkeavuuksien tunnistamiseksi täytyy tarkastella muitakin muuttujia. Peilatessa sovittua vuosikorkoa markkina-arvoon huomataan, että koron suhteen ääriarvoja löytyy ainoastaan markkina-arvon ollessa lähellä nollaa. Tämä on todennäköisesti seurausta siitä, että korkojen ääriarvot ovat kokolailla poikkeuksetta virheellisiä, joten ne pyritään korjaamaan aina.



Kuva 9 Poikkeavuusluokan jakaantuminen aineistossa (akseleiden asteikot peitetty aineiston sensitiivisyyden vuoksi)

4.2.4 Lopullinen tutkimusaineisto

Aineiston alkuperäinen 91 muuttujan joukko redusoidaan datahaun jälkeen 83 dimension joukkoon. Tämä 83 dimension joukko esikäsitellään puuttuvien havaintojen sekä kategoristen muuttujien osalta. Osa puuttuvia havaintoja sisältävistä muuttujista muokataan uusiksi binäärisiksi muuttujiksi ja samalla poistetaan alkuperäiset muuttujat, joten puuttuvien havaintojen käsittely ei vaikuta muuttujien lukumäärään. Yhteensä 17 kategorista muuttujaa käsitellään one-hot-koodauksella, jonka tuloksena lopulliseksi tutkimusaineistoksi muodostuu 165:n muuttujan joukko. Tämän jälkeen aineistosta poistetaan vielä ”kuolleet” muuttujat eli sel-

laiset, joiden arvo on vakio. Tällaisilla muuttujilla ei ole luokittelun kannalta arvoa ja lisäksi kuolleiden muuttujien nollavarianssi aiheuttaa laskentaongelmia joidenkin menetelmien kohdalla, joten näitä ei ole syytä säilyttää tutkimusaineistossa. Lisäksi kaksi muuttujaa käytetään rivien nimeämiseen. Muuttuja CALC_Flow_Id on rivin identifioiva tunniste, joka on koostettu Suomen Pankissa. Tähän tunnisteeseen yhdistetään raportoijan nimi, sillä on mahdollista, että useamman raportoijan raporteilla esiintyy sama CALC_Flow_Id, koska tunnus ei sisällä raportoijatietoa. Tunnisteen muodostamisen jälkeen rivit nimetään tunnisteen mukaan ja kaikki kolme muuttujaa (CALC_Flow_ID, raportoijan nimi, tunniste) poistetaan.

Havaintojen lukumäärä aineistossa on n. 450 000, kun mukana on valitun periodin ensimmäiset raportit. Aineistoa on mahdollista laajentaa käytännössä kuinka suureksi tahansa, mutta laskentakapasiteetin rajallisuuden vuoksi päädyttiin käyttämään perusaineistona yhden periodin havaintoja. Poikkeavien havaintojen osuus koko aineistossa on 14 %.

5. MALLIEN SOVITUS

Tässä luvussa kartoitetaan luvussa neljä esiteltyjen koneoppimismenetelmien kyky tunnistaa poikkeavia havaintoja tutkimusaineistosta. Tavoitteena on löytää menetelmä, joka onnistuu parhaiten tunnistamaan poikkeavuudet normaaleista havainnoista. Toissijaisena mielenkiinnon kohteena on selvittää, auttavatko valitut menetelmät hahmottamaan aineiston rakennetta ja säännönmukaisuuksia.

5.1 K-means

Aineiston muuttujat kuvaavat monipuolisesti rahalaitosten tase-erien eri ominaisuuksia ja näin ollen muuttujien asteikot vaihtelevat suuresti toisiinsa nähden, kun toiset muuttujat kuvaavat korkoa ja toiset markkina-arvoja tai esimerkiksi vastapuolen tietoja. Euromääräiset muuttujat, joiden varianssi voi olla hyvinkin suuri, eroavat merkittävästi esimerkiksi dummy-muuttujiksi käännetyistä kategorisista muuttujista, joiden varianssi sijoittuu välille [0,1]. Datajoukon alkioiden etäisyydet lasketaan tässä tapauksessa 165-dimensionaalisessa avaruudessa jokaisen dimension suhteen, joten suuren varianssin omaavien muuttujien vaikutus jäännösneliövirheeseen olisi merkittävämpi kuin muiden muuttujien ja näin ollen niiden paino klusteroinnissa olisi suurempi kuin pienen varianssin omaavien muuttujien. Siispä aineiston muuttujat tulee joko standardoida tai normalisoida ennen klusterointia, jotta tiettyjä dimensioita ei lähtökohtaisesti painotettaisi liikaa klusteroinnin näkökulmasta. Muuttujien standardointi keskiarvon ja varianssin pohjalta vaatii muuttujien normaalijakautuneisuuden, joka ei tutkimusaineistossa päde. Standardoinnin sijaan muuttujat voidaan normalisoida välille [0,1]. Havainto x_i normalisoidaan muuttuja j suhteen kaavan 23 mukaisesti.

$$x_{ij}^{Norm} = \frac{x_{ij} - min_{x_j}}{max_{x_j} - min_{x_j}}, i = 1, \dots, n, missä \qquad (23)$$

$n = havaintojen\ lukumäärä$

Kategoristen muuttujien dummy-muuttujia ei luonnollisesti tarvitse normalisoida erikseen, koska normalisointi ei vaikuttaisi muuttujien arvoon. Taulukko 1 havainnollistaa normalisoinnin vaikutusta aineiston variansseihin.

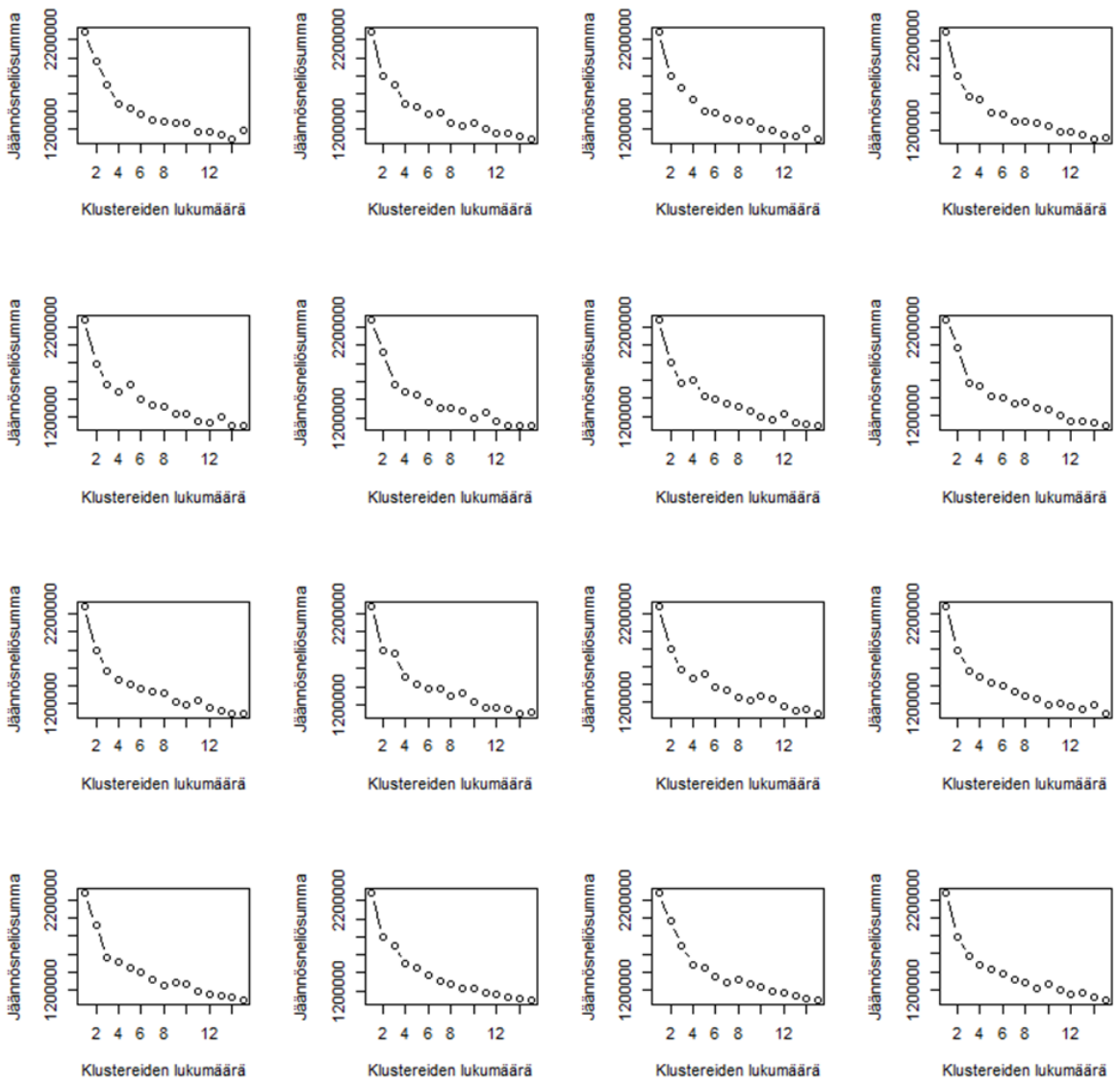
Taulukko 1 Varianssien tunnuslukuja ennen ja jälkeen normalisoinnin

	Varianssien tunnusluvut	
	Alkuperäinen aineisto	**Normalisoitu aineisto**
Minimi	2.2e-06	1.7e-06
Maksimi	6.1e+20	0.3
Keskiarvo	4.7e+18	0.04
Mediaani	0.02	0.002

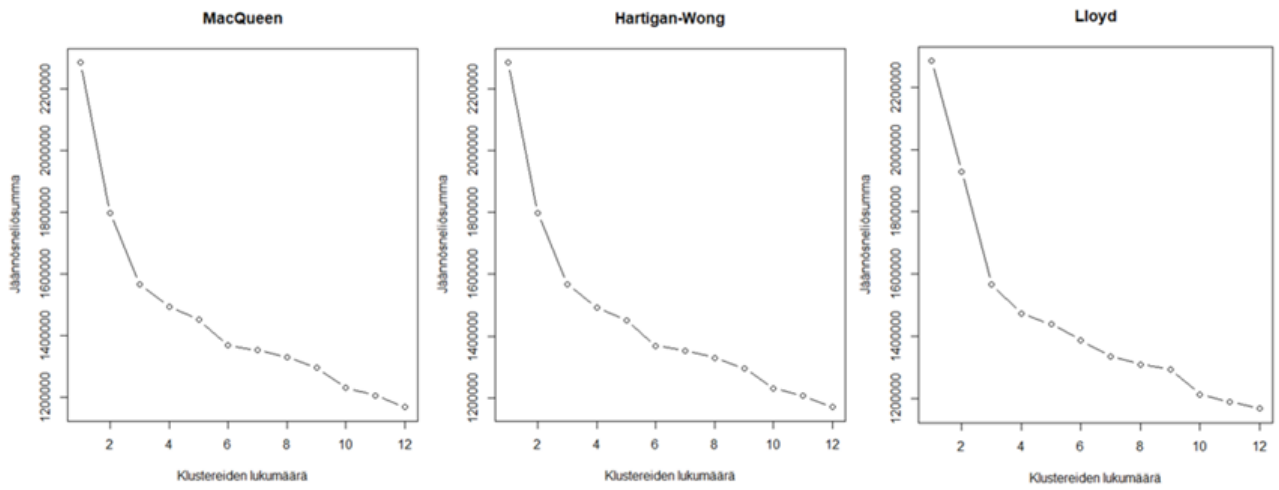
5.1.1 Klustereiden lukumäärä

Klustereiden optimaalisen lukumäärän määrittämiseen ei ole olemassa varsinaista estimointimenetelmää, vaan sopiva K:n arvo löydetään kokeilemalla. Sopivan K-arvon valinnassa voidaan käyttää apuna niin kutsuttua ”kyynärpää”-kuvaajaa (elbow graph). Kyynärpääkuvaajan y-akseli edustaa ryhmien sisäistä jäännösneliösummaa ja x-akseli klustereiden lukumäärää. Mitä jyrkemmin piirretty käyrä laskee y-akselin suuntaisesti, sitä suurempi vaikutus klusterin lisäämisellä on neliösumman pienentymiseen. Pisteen, jossa käyrän lasku taittuu tasaiseksi siten, ettei neliösumma enää merkittävästi pienene k:n kasvaessa, katsotaan olevan optimaalisin k:n arvo. Graafisesti tarkasteltuna tämä taite näkyy kuvaajassa kyynärpään muotoisena.

Lähtökeskipisteen valinnasta riippuen merkittävin lasku jäännösneliösummassa on klusterin 3 tai 4 lisäyksen jälkeen. Kun lähtökeskipiste valitaan satunnaisesti 100 kertaa Lloydin algoritmille, osoittautuu graafisen tarkastelun perusteella klustereiden optimaaliseksi lukumääräksi neljä (kuva 10). Kun luokittelu tehdään kaikilla luvussa 3.3.1 kuvatuilla algoritmeilla huomataan, että optimaaliseksi k:n arvoksi valikoituu jokaisen algoritmin tapauksessa k = 4 (kuva 11). Eri algoritmien tuottamissa jäännösneliösummissa ei ole juurikaan eroja, joskin Lloydin algoritmin tuloksissa on pienoinen ero kahteen muuhun algoritmiin verrattuna. Verrattuna MacQueenin algoritmiin Hartigan-Wong –algoritmin tuloksista voidaan päätellä, että datajoukossa ei ole alkioita, jotka olisivat sijoitettu muuhun kuin lähimpänä sijaitsevaan klusteriin. Lopullisessa mallissa käytetään neljää klusteria sekä Lloydin algoritmia.



Kuva 10 Lloydin klustereiden jäännösneliösummat eri iterointikierroksilla

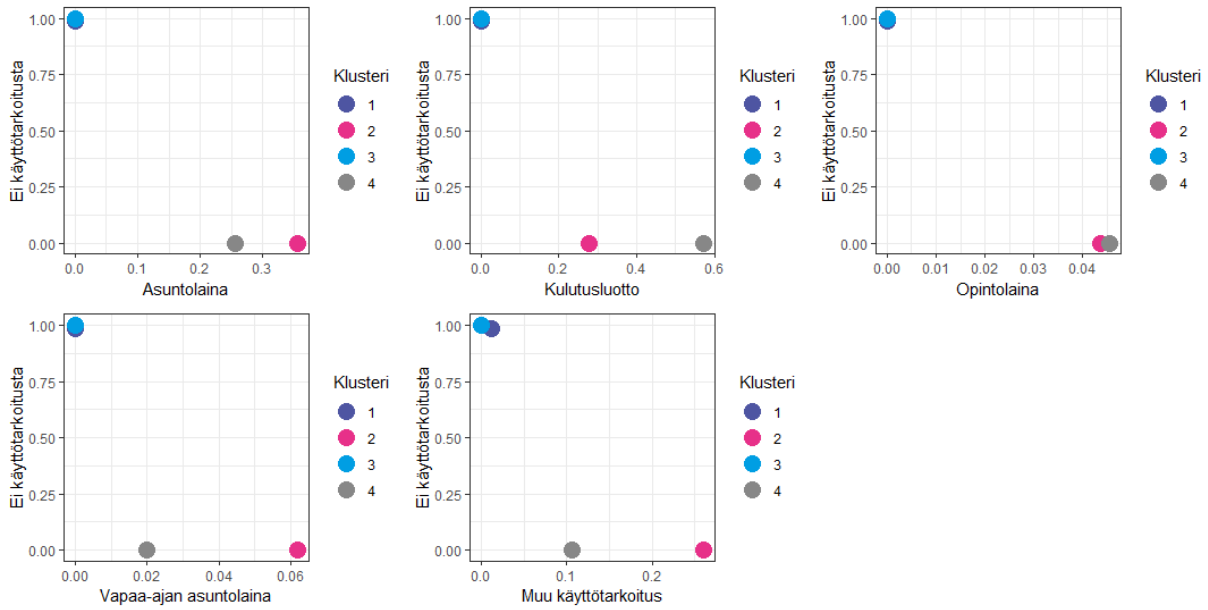


Kuva 11 K:n lukumääräksi valikoituu neljä kaikilla algoritmeilla

5.1.2 Klustereiden tulkinta

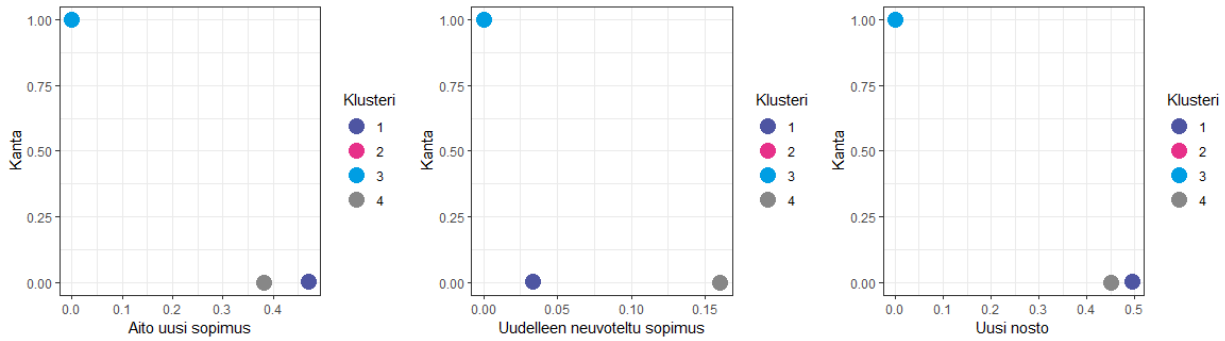
K-means-klusteroinnissa tarkoituksena on löytää aineistosta säännönmukaisuuksia, joiden avulla datapisteitä voidaan luokitella mahdollisimman samankaltaisia havaintoja sisältäviin luokkiin. Graafisen tarkastelun tuloksena tutkimusaineiston jakaminen neljään luokkaan vaikuttaisi erottelevan datapisteet tehokkaasti siten, että datapisteiden väliset erot ovat mahdollisimman pienet kunkin klusterin sisällä ja toisaalta suuret klustereiden välillä. Klusteroinnin tulosten visualisointi ei ole suoraviivaista, kun kyseessä on lähes puolen miljoonan datapisteen ja 165 muuttujan aineisto. Aineiston muuttujien normalisoidut arvot aggregoidaan klustereittain siten, että jokaiselle klusterille lasketaan muuttujakohtaiset keskiarvot. Muuttujakeskiarvoja tarkastelemalla saadaan käsitys siitä, miten ja mitkä klusterit kuvaavat mitäkin muuttujia. Osalla muuttujista keskiarvot ovat hyvin samaa luokkaa kussakin klusterissa, eli muuttujien arvot eivät keskimääräisesti juurikaan eroa klustereiden välillä. Muuttujat, joiden keskiarvot eroavat selvästi klustereiden välillä ovat etenkin lainan käyttötarkoitusta, taloustoimea, maturiteettia sekä vakuutta kuvaavat dummy-muuttujat.

Kun klustereita tarkastellaan käyttötarkoituksittain (kuva 12) huomataan, että klusterit 1 ja 3 ovat hyvin samankaltaisia näiden muuttujien osalta. Molemmat klusterit sisältävät lähes ainoastaan lainoja, joille käyttötarkoitusta ei ole määritelty. Sen sijaan klusterit 2 ja 4 sisältävät kaikkia muita käyttötarkoituksia – klusteri 4 erityisesti kulutusluottoja ja klusteri 2 muun käyttötarkoituksen lainoja.



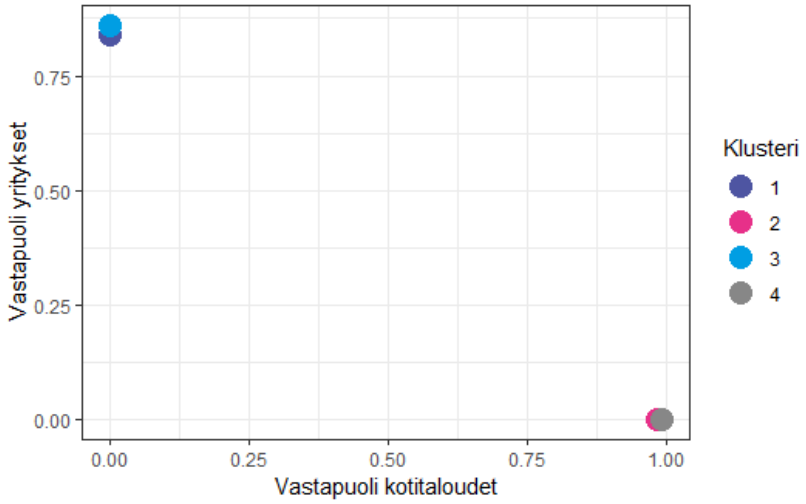
Kuva 12 Klusterit käyttötarkoituksen mukaan jaoteltuna

Taloustoimen mukaan luokiteltuna (kuva 13) klusterit 2 ja 3 kuvaavat kantaa, kun taas klusterit 1 ja 4 sisältävät muita taloustoimia. Klustereita 2 ja 3 ei pysty erottamaan toisistaan taloustoimen perusteella, sillä molemmat klusterit sisältävät ainoastaan lainojen kantaa kuvaavia datapisteitä.



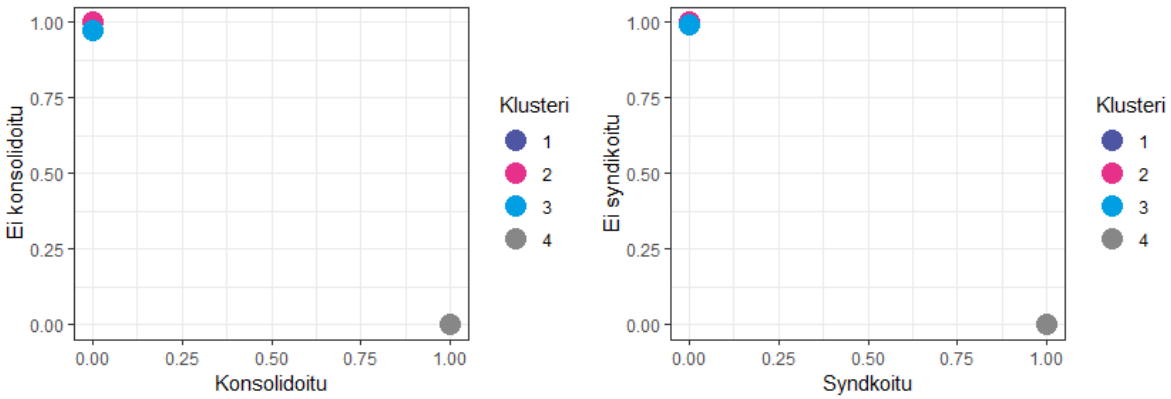
Kuva 13 Klusterit taloustoimen mukaan jaoteltuna

Vastapuolen sektoreista kotitaloudet ja yritykset jakavat klusterit selvästi kahteen luokkaan. Klusteri 1 ja 3 sisältävät yrityssektorin, kun taas klusterit 2 ja 4 sisältävät kotitaloussektorin (kuva 14). Tulos on odotettu, sillä yritys- ja kotitalouslainat oletusarvoisesti eroavat ominaisuuksiltaan toisistaan.



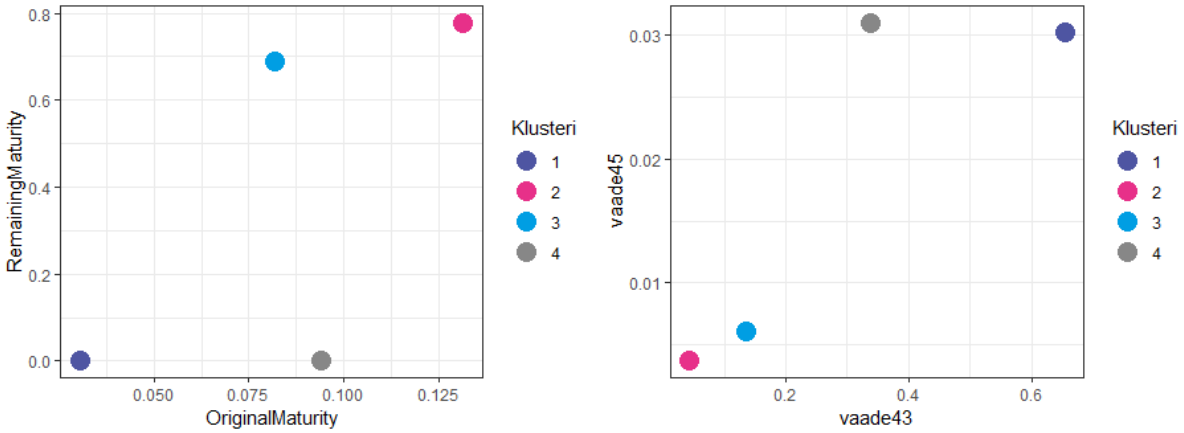
Kuva 14 Klusterit vastapuolen sektorin mukaan jaoteltuna

Klusterit voidaan jakaa kahteen selvästi erilliseen luokkaan myös konsolidoinnin sekä syndikoinnin perusteella. Lainat, joita ei ole syndikoitu tai konsolidoitu sijaitsevat klustereissa 2 ja 3 (kuva 15). Tämä tulos on jokseenkin yllättävä, koska näillä vaateilla ei ole ajateltu olevan kovin suurta merkitystä havaintojen luokittelun kannalta.



Kuva 15 Klusterit vakuuden mukaan jaoteltuna

Lisäksi klustereiden väliset keskiarvot eroavat toisistaan etenkin maturiteettimuuttujien ja joidenkin vaateiden mukaan. Kuvasta 16 nähdään, että klusterit 2 ja 3 sisältävät lainoja, joilla jäljellä oleva maturiteetti on pitkä, kun taas klustereissa 1 ja 4 olevilla havainnoilla ei ole jäljellä olevaa maturiteettia. Klusterissa 2 on kaikkein pisimmät sekä alkuperäiset että jäljellä olevat maturiteetit. Keskimmäisestä kuvaajasta havaitaan, että klusteri 1 sisältää vaadetta 43 selvästi enemmän kuin muut klusterit.



Kuva 16 Klusterit maturiteetin ja vakuuden mukaan jaoteltuna

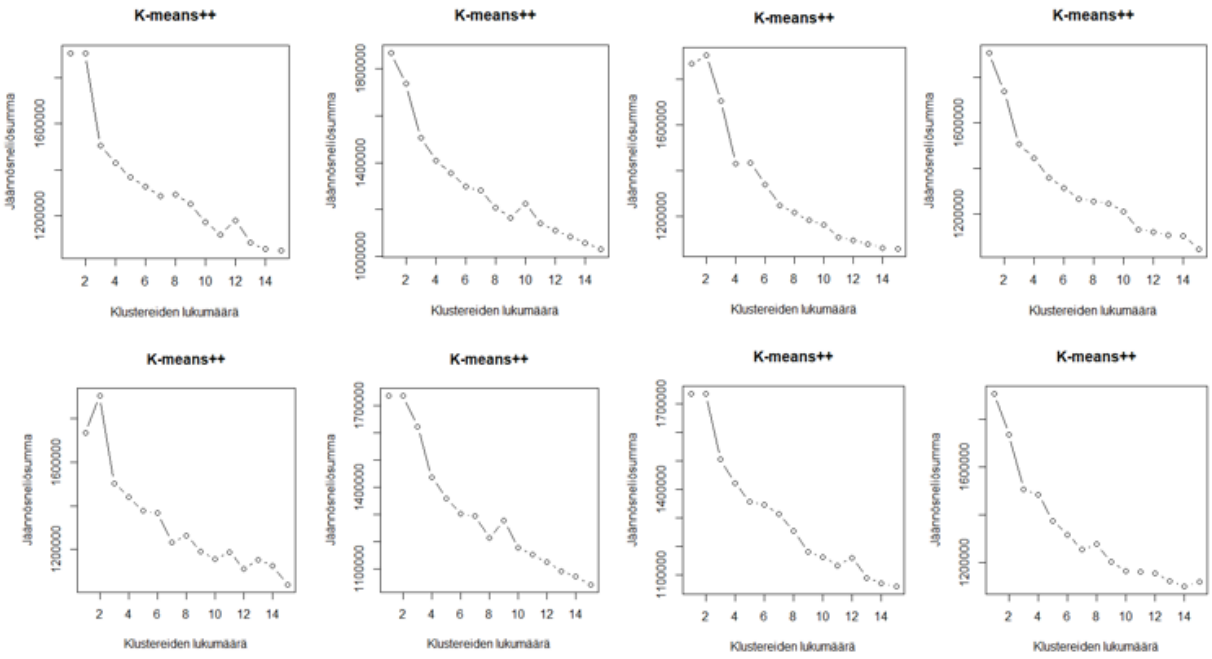
K-means-klusteroinnin tuloksena saadut klusterit jaottelevat tutkimusaineiston havainnot selvästi toisistaan eroaviin klustereihin. Ensimmäinen klusteri kuvaa havaintoja, joiden vaadeluokka on jokin muu kuin tililuotot (43) tai luottokorttien maksuaikaluotot (45), joilla on pitkä maturiteetti, taloustoimi on kanta ja lainat ovat vakuudettomia (vakuus U). Klusteri 1 sisältää myös yrityslainoja ja lainoja, jotka ovat syndikoituja ja konsolidoituja. Klusteri 2 kuvaa niin ikään pitkän maturiteetin lainoja, joiden vaade on muu kuin 43 ja 45, mutta vakuutena on asunto- ja kiinteistövakuus (M) ja vastapuolen sektorina kotitaloudet. Klusteri 3 on vaateiden, maturiteettien ja taloustoimen osalta samankaltainen kuin klusteri 2, mutta on vakuuksien osalta heterogeenisempi kuin klusteri 2 ja suurimpana erona kuvaavin vastapuolen sektori on yritykset. Klusteri 4 puolestaan sisältää lainoja, joilla ei ole jäljellä olevaa maturiteettia, mutta alkuperäinen maturiteetti on ollut pitkä. Klusteri on heterogeeninen sekä käyttötarkoituksen että taloustoimen osalta ja sisältää erityisesti konsolidoituja ja syndikoituja lainoja kotitalouksille. Poikkeavat havainnot ovat jakautuneet tasaisesti kaikkiin klustereihin (taulukko 2), joten vaikkakin havainnot saatiin luokiteltua selkeästi erilaisiin klustereihin, täytyy poikkeavuuksien tunnistamiseksi harkita muita menetelmiä.

Taulukko 2 poikkeavien havaintojen osuudet klustereittain

	Poikkeavien havaintojen osuus
Klusteri1	14 %
Klusteri2	14 %
Klusteri3	14 %
Klusteri4	14 %

5.2 K-means++

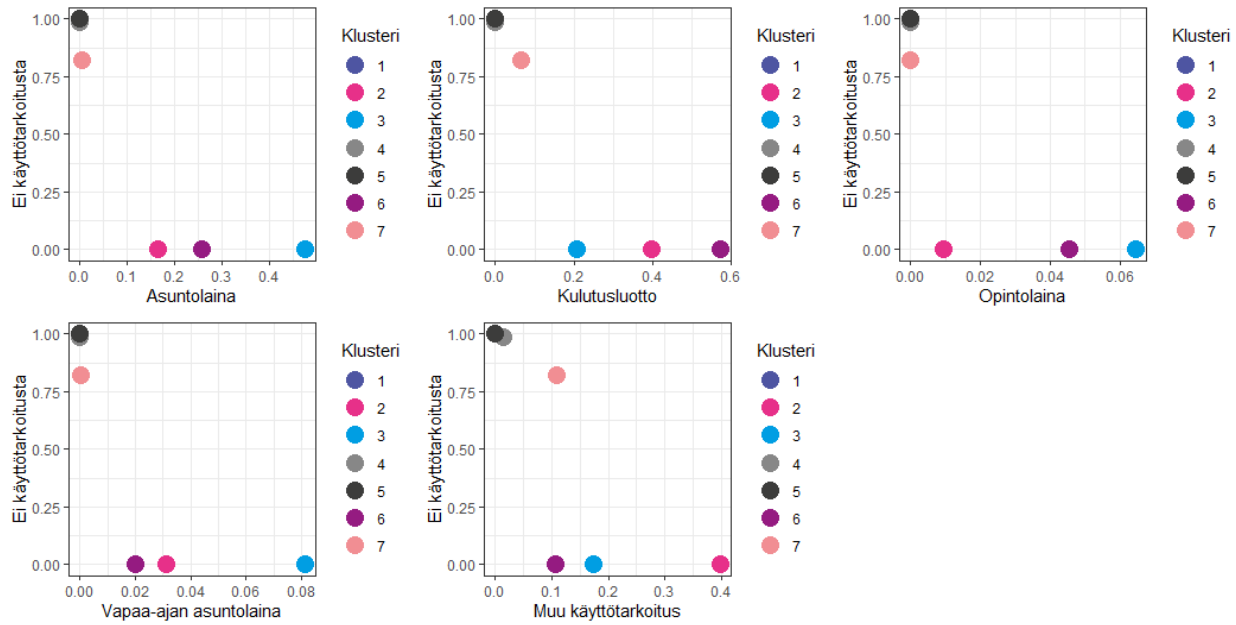
K-means++-menetelmällä muodostettavien klustereiden lukumääräksi valikoitui seitsemän. Kun iterointikierroksia suoritettiin 100, keskimääräisesti kyynärpääkuvaajan taitekohta asettui seitsemännen klusterin kohdalle (kuva 17).



Kuva 17 K-means++ jäännösneliösummat eri iterointikierroksilla

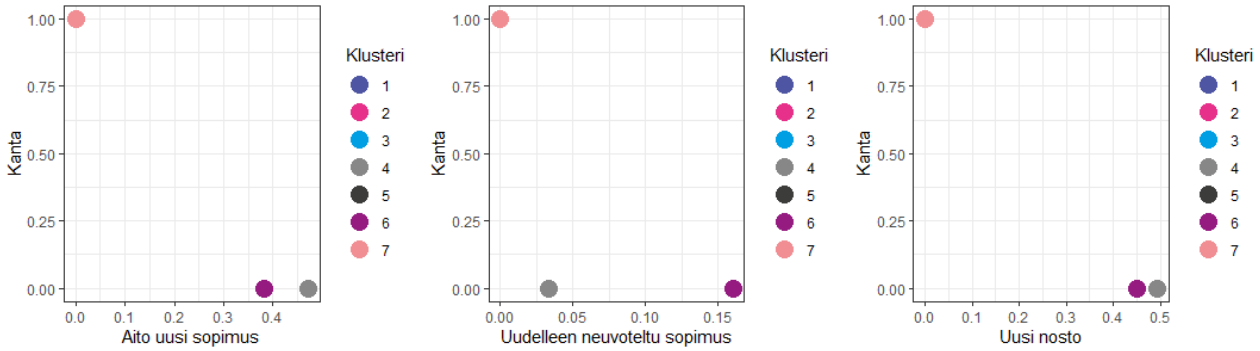
Kun muodostettuja klustereita tarkastellaan muuttujapareittain huomataan, että muuttujien arvojen jakautuminen klustereittain on hyvin samankaltainen kuin K-means-klusteroinnissa. Lainatyypin mukaan voidaan erottaa selvästi toisistaan klusterit, jotka sisältävät lainoja joille käyttötarkoitusta ei ole määritelty klustereista, jotka sisältävät eri käyttötarkoitusten lainoja. Kuvasta 18 nähdään, että klustereihin 1,4 ja 5 on asetettu datapisteet, joille käyttötarkoitusta ei ole määritelty. Klusterit 1 ja 5 ovat identtisiä lainatyypin suhteen, sillä molemmat sisältävät ainoastaan tyyppiä ”Ei käyttötarkoitusta”. Myös klusteri 7 sisältää pääasiassa lainoja, joilla käyttötarkoitusta ei ole, mutta näiden lisäksi klusterissa on havaintoja, joiden käyttötarkoitus on ”Muu” tai ”Kulutusluotto”. Vapaa-ajan asuntolainat on asetettu pääasiassa klusteriin 3, joka pääasiallisesti

sisältää asuntolainoja. Klusteri on kuitenkin melko heterogeeninen lainatyypin suhteen sisältäen myös kolmea muuta lainatyyppiä. Klusteri 2 sisältää pääsääntöisesti muun käyttötarkoituksen lainoja sekä kulutusluottoja ja klusteri 6 opintolainoja sekä kulutusluottoja.



Kuva 18 Klusterit käyttötarkoituksen mukaan jaoteltuna

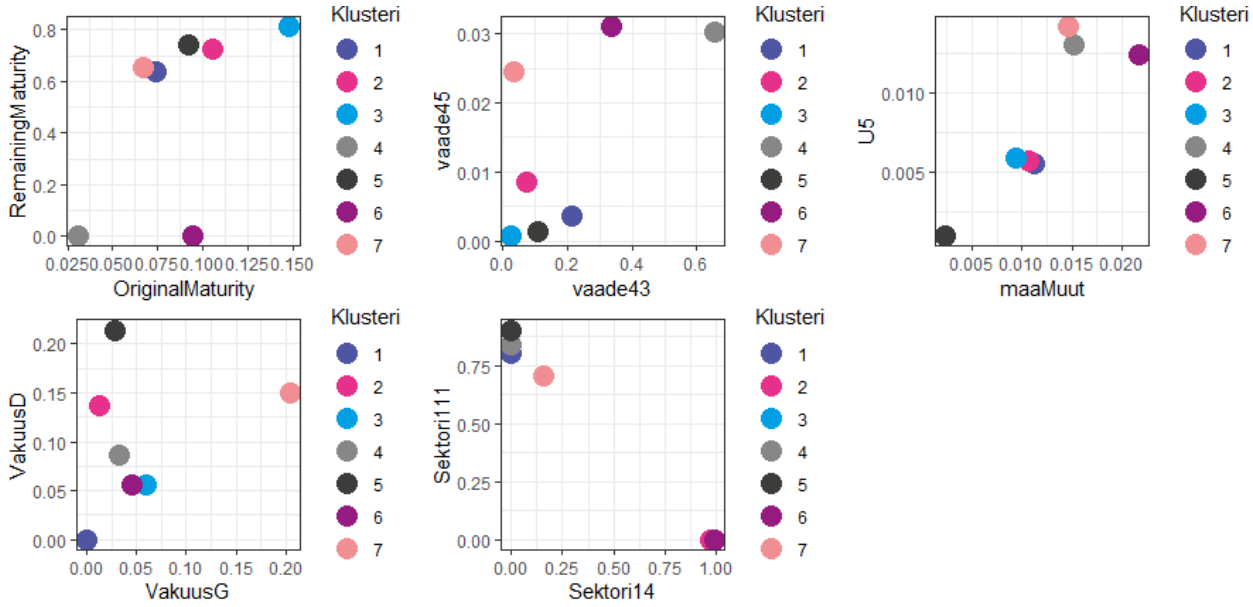
Tarkasteltaessa klustereita taloustoimen mukaan huomataan, että klusterit ovat täysin identtisiä verrattuna K-means-menetelmällä luotuihin klustereihin (kuva 19). Voidaan siis päätellä, ettei K-means++ tuota lisäarvoa klusterointiin taloustoimen näkökulmasta.



Kuva 19 Taloustoimen osalta k-means++ -klusterit ovat samanlaisia kuin k-means -klusterit

Lainatyypin tai taloustoimen avulla ei pystytty tunnistamaan seitsemää erillistä klusteria, joka kuitenkin virheneliösumman perusteella on optimaalisin klusterilukumäärä. Kun tarkastelun alle otetaan maturiteettiluokka, vaadeluokka, vakuus sekä vastapuolen maaryhmä, pystytään kaikki seitsemän klusteria tunnistamaan

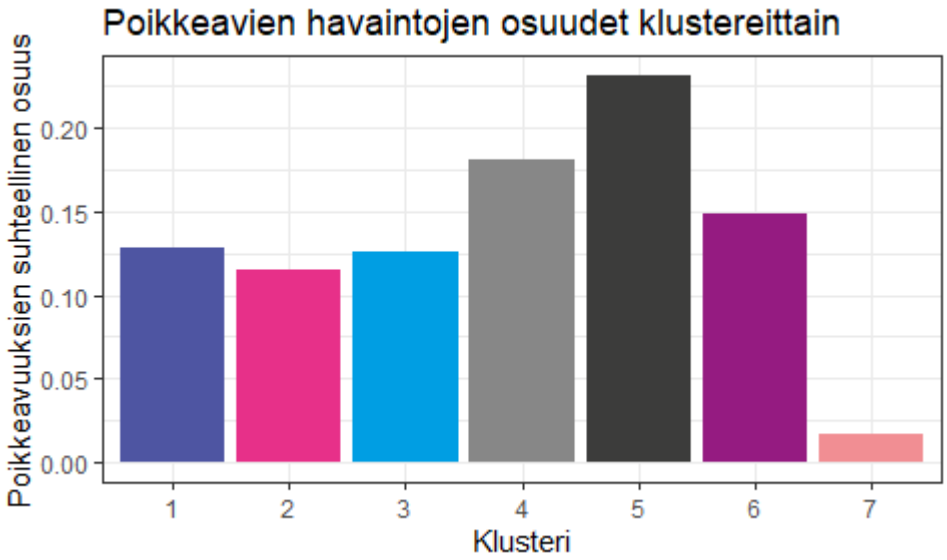
(kuva 20). Alkuperäisen ja jäljellä olevan maturiteetin mukaan klusterit 3 ja 4 ovat kaikkein kauimpana toisistaan klusterin 3 sisältäessä havaintoja, joilla sekä alkuperäinen että jäljellä oleva maturiteetti ovat pitkiä ja klusterin 4 sisältäessä havaintoja, joilla ei ole jäljellä olevaa maturiteettia ja alkuperäinenkin maturiteetti on lyhyt. Muut klusterit sijoittuvat maturiteettien mukaan tarkasteltuna näiden kahden klusterin välimaastoon. Vaateen mukaan katsottuna klusteri 4 ja 6 sisältävät selvästi suuremman osuuden vaadeluokkaa 43 ja 45 verrattuna muihin klustereihin. Klusteri 3 sisältää muita vaateita kuin 43 tai 45, ja muut klusterit sisältävät jonkin verran näitä vaateita, mutta kuitenkin suurelta osin muita vaadeluokkia. Vastapuolen sektorin mukaan klusterit jakautuvat niihin, jotka sisältävät sektoria ”Kotitaloudet” (Sektori14, klusterit 1 ja 6) sekä niihin, jotka sisältävät sektoria ”Yritykset” (Sektori111). Näistä hieman erillään sijaitsee klusteri 7, jonka enemmistösektori on yritykset, mutta sisältää myös kotitaloussektoria sekä muita sektoreita jonkin verran.



Kuva 20 Klusterit erottuvat toisistaan parhaiten vaateen mukaan tarkasteltuna

Poikkeavat havainnot jakautuvat K-means++-klustereihin epätasaisemmin kuin K-means-klustereihin. Kuvasta 21 nähdään, että klusteri 5 sisältää selvästi eniten poikkeavia havaintoja 23 % suhteellisella osuudella. Klusteri 7 sen sijaan koostuu lähes täysin normaaleista datapisteistä. Havainnosta voidaan päätellä, että poikkeavia havaintoja on erityisesti sellaisten lainojen joukossa, joille ei ole määritelty käyttötarkoitusta. Klusterin 5 ominaispiirteenä on myös vakuusluokka ”D” sekä vastapuolen sektori ”Yritykset”, lisäksi vastapuolen maaryhmänä on jokin muu kuin ”U5” tai ”Muut”. Vähiten poikkeavuuksia sisältävä klusteri on vastapuolen sektorin osalta kaikkein heterogeenisin, mutta sisältää muita klustereita enemmän vakuuden ”G” sekä vastapuolen maaryhmän ”U5” datapisteitä. Niin ikään kaikki klusterin 7 datapisteistä edustavat jotain muuta taloustoimea kuin kantaa. Tuloksesta voidaan päätellä, että uusien lainojen (uusi nosto/uudelleen neuvoteltu

sopimus/aito uusi sopimus) tapauksessa poikkeavia havaintoja ei tunnisteta niin herkästi laadunvalvontaprosessissa.



Kuva 21 Klusteri 5 sisältää eniten poikkeavia havaintoja, kun taas klusteri 7 koostuu pääasiassa normaaleista havainnoista

5.3 K-medoids

K-medoids on K-means- ja K-means++-menetelmien sukulainen, jossa klustereiden keskipiste määritetään keskiarvon sijaan alkioiden erilaisuuden perusteella. Kuten sukulaismenetelmät, myös K-medoids ottaa syöteparametrinaan klustereiden lukumäärän.

5.3.1 Klustereiden lukumäärä

Klustereiden lukumäärän valinnassa käytetään apuna silhuettiarvoa, joka kuvaa datapisteiden keskimääräistä etäisyyttä klusterin sisällä suhteessa datapisteiden keskimääräiseen etäisyyteen lähimmän klusterin datapisteisiin. R-ohjelmiston kmed-paketin sil-funktiolla lasketaan siluettiarvo kaavan 24 mukaisesti

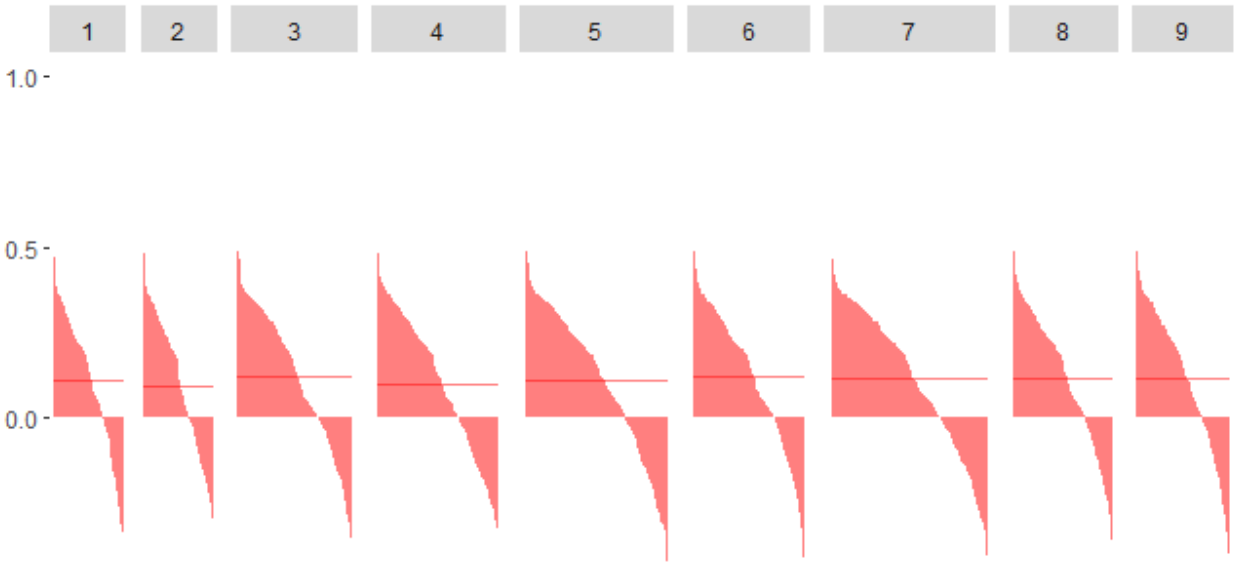
$$s_i(i) = \frac{b_i - a_i}{\max(a_i, b_i)}, missä \qquad (24)$$

$a_i = Datapisteen\ i\ keskimääräinen\ etäisyys\ klusterin\ muihin\ datapisteisiin\ ja$

$b_i = Datapisteen\ i\ keskimääräinen\ etäisyys\ lähimmän\ klusterin\ datapisteisiin$

Mitä suurempi klusterin siluettiarvojen keskiarvo on, sitä onnistuneempana klusterointia voidaan pitää. Negatiiviset siluettiarvot viittaavat siihen, että kyseinen datapiste on asetettu klusteriin, jonka datapisteisiin

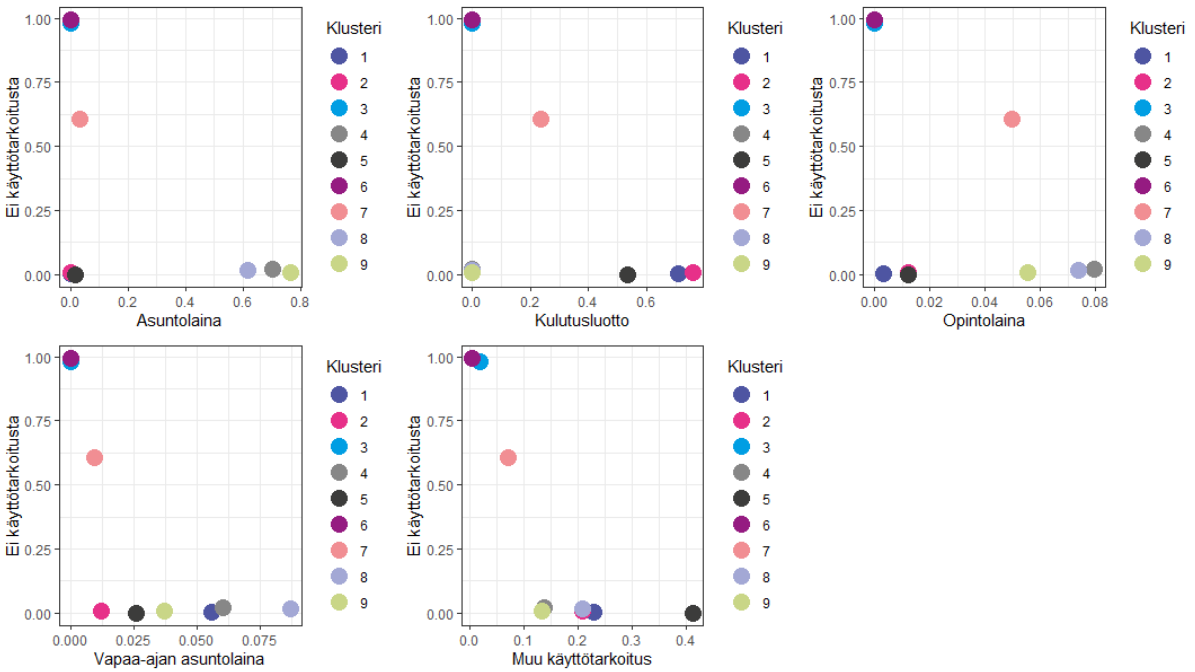
keskimääräinen etäisyys on suurempi kuin lähimmän klusterin datapisteisiin. Kun suoritetaan sata iterointikierroksia klustereiden eri lukumäärille, valitaan klustereiden lukumääräksi yhdeksän. Kuva 22 esittää siluettiarvot klustereittain.



Kuva 22 K-medoids siluettikuvaaja muodostetuille klustereille

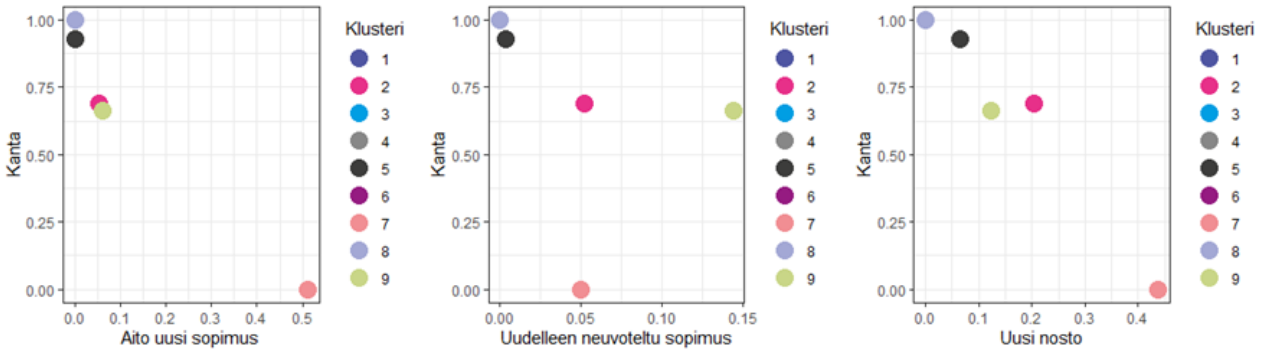
5.3.2 Klustereiden tulkinta

Lainan käyttötarkoituksen mukaan jaoteltuna (kuva 23) huomataan, että klusterit 1, 2, 5, 4, 8 ja 9 ovat sellaisia, jotka eivät sisällä lainoja, joille käyttötarkoitusta ei olisi määritelty. Sen sijaan klusterit 3 ja 6 ovat lähes päällekkäisiä käyttötarkoituksen suhteen ja koostuvat lainoista, joille käyttötarkoitusta ei ole määritelty. Klusteri 7 erottuu selvästi muista klustereista sisältäen kaikkia käyttötarkoitusluokkia.



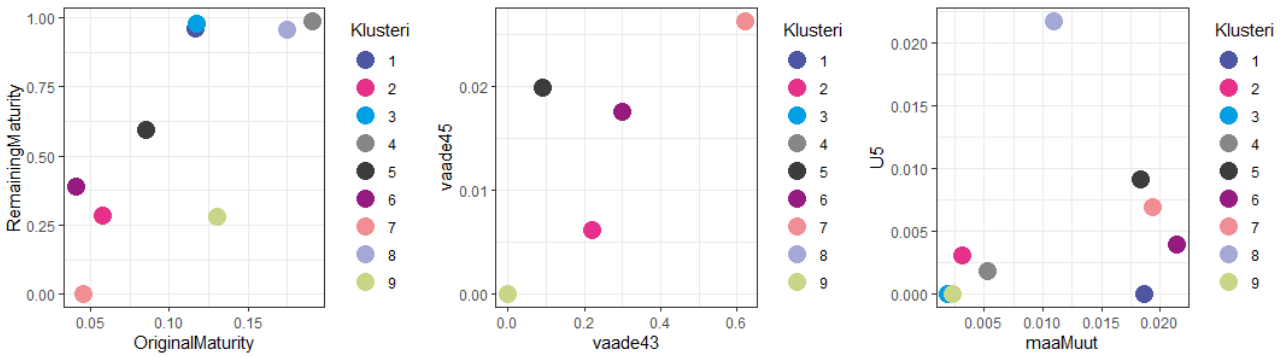
Kuva 23 K-medoids-klusterit lainan käyttötarkoituksen mukaan jaoteltuna

Taloustoimen perusteella jaoteltuna on tunnistettavissa viisi erillistä ryhmää klustereiden 1, 3, 4, 6 ja 8 ollessa päällekkäisiä taloustoimen suhteen. Klusteri 7 erottuu selvästi muista, sillä se ei sisällä lainkaan taloustoimen ”kanta” omaavia datapisteitä, vaan koostuu lähinnä uusista nostoista sekä aidoista uusista sopimuksista. Uudelleen neuvoteltuja sopimuksia löytyy pääasiassa klusterista 9 (kuva 24).



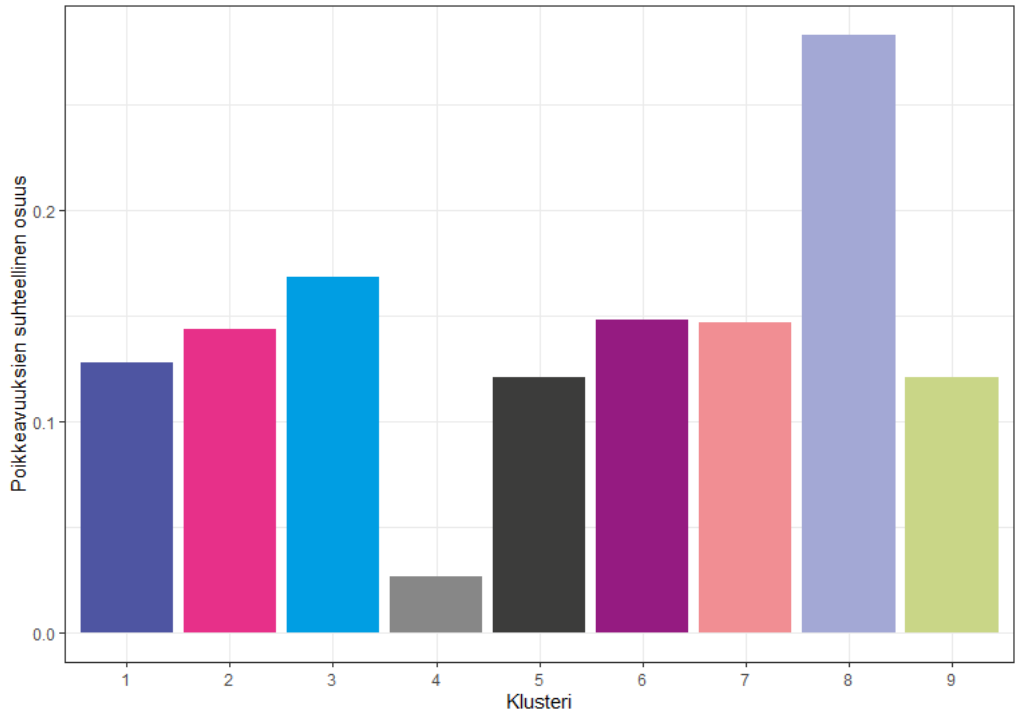
Kuva 24 K-medoids-klusterit taloustoimen mukaan jaoteltuna

Yhdeksän klusteria erottuvat toisistaan maaryhmä- ja maturiteettijaottelulla, kun taas vaateen mukaan jaoteltuna löytyy päällekkäisiä klustereita (kuva 25). Maaryhmän mukaan jaoteltuna klusterit 2,3,4 ja 9 muistuttavat toisiaan sisältäen pääasiassa muita ryhmiä kuin ”maaMuut” ja ”U5”. Klusteri 8 taas sisältää erityisesti U5-ryhmän datapisteitä, ja loput klustereista edustavat ”maaMuut”-ryhmää. Vaateen osalta klusterit 1,3,4,8 ja 9 ovat päällekkäisiä sisältäen muita vaadeluokkia kuin 43 ja 45.

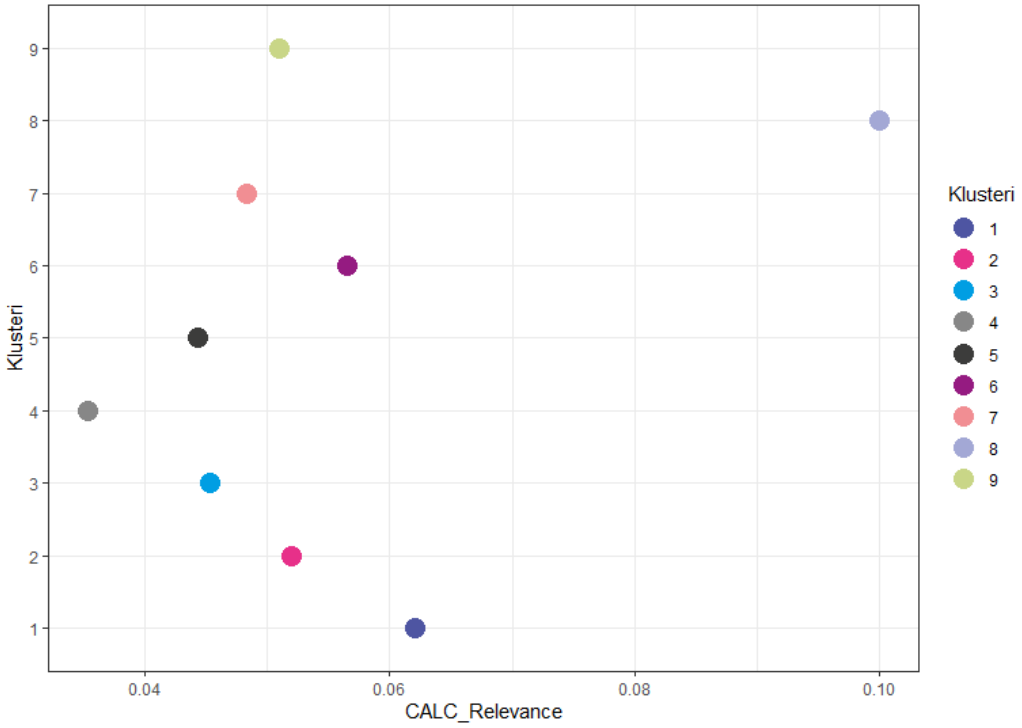


Kuva 25 K-medoids-klusterit maturiteetin, vaateen ja maaryhmän mukaan jaoteltuna

Poikkeukselliset havainnot ovat jakautuneet melko tasaisesti klustereiden välillä (kuva 26). Kuitenkin samoin kuin K-means++-klusteroinnin tapauksessa, kaksi klusteria erottuvat muusta joukosta poikkeavuuksien esiintymistiheyden suhteen. Klusterissa 8 on selvästi suurempi osuus poikkeavia havaintoja kuin muissa klustereissa, joissa poikkeavuuksien havaintojen osuus on lähellä poikkeavuuksien osuutta koko aineistossa. Klusteri 4 erottuu myös muusta joukosta sen sisältäessä erityisen vähän poikkeavia havaintoja. Yllä tarkastelluista muuttujista maaryhmä on ainoa, jonka suhteen klusterit 4 ja 8 selvästi eroavat toisistaan. Havainto herättää epäilyksen, että muuttujatarkastelu ei ole ollut kyllin kattava. Muuttujien uudella läpikäynnillä huomataan, että CALC_Relevance on sellainen muuttuja, jonka suhteen klusterit 4 ja 8 eroavat toisistaan selvästi. Muuttuja on Suomen Pankissa raportille rikastettu kenttä, joka kuvaa havainnon aggregointitasoa eli sitä, onko havainto (rivi) summattu useammasta pohjarivistä. Mikäli rivi on muodostettu pohjariveistä, on nämä summaukseen käytetyt pohjarivit poistettu aineistosta, joten tällaiset havainnot eivät kuitenkaan esiinny kahteen kertaan aineistossa. Kuvasta 27 nähdään, että klusterit 4 ja 8 ovat kauimpana toisistaan CALC_Relevance-muuttujan suhteen. Aggregoimaton datapiste saa muuttujalle pienemmän arvon, joten havainto viittaa siihen, että aggregoiduilla summariveillä esiintyy enemmän poikkeavuuksia.



Kuva 26 Poikkeavien havaintojen suhteelliset osuudet klustereittain



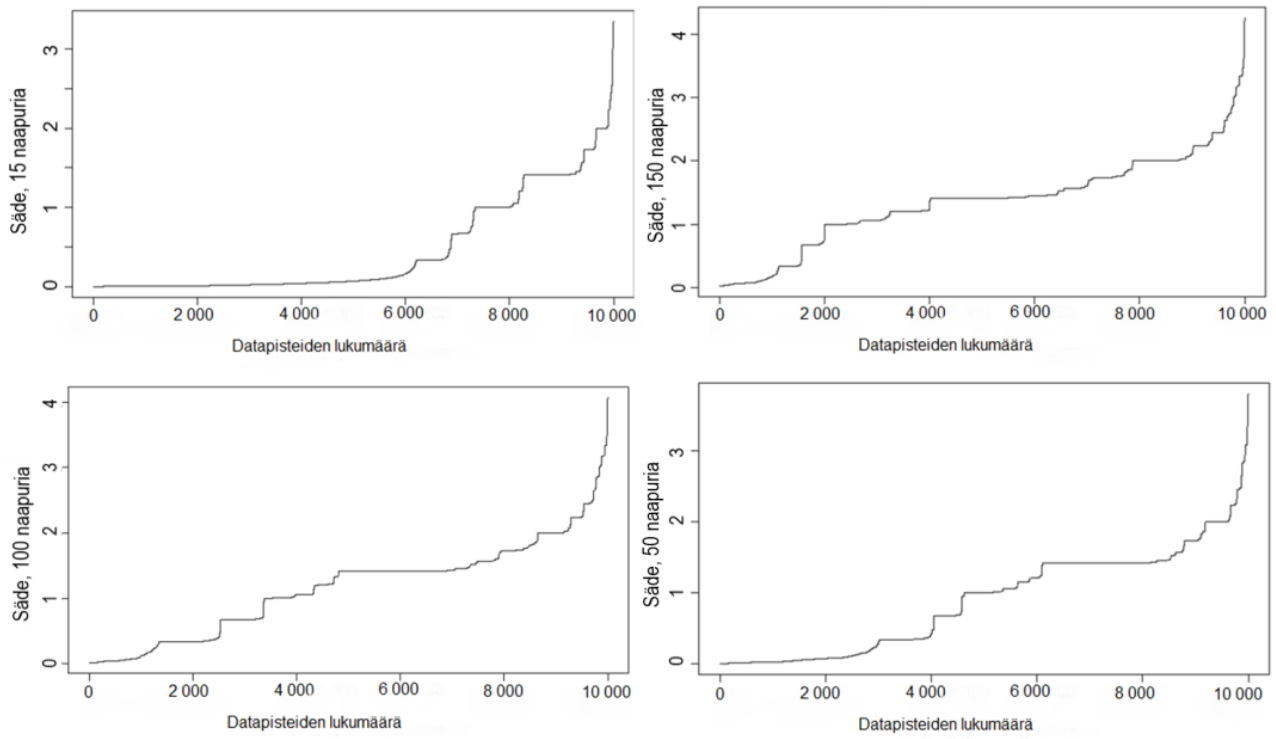
Kuva 27 CALC_Relevance on tärkein muuttuja erottamaan klusterit 4 ja 8 toisistaan

5.4 DBSCAN

DBSCAN on K-means-pohjaisten menetelmien tapaan osittava ohjaamattoman oppimisen menetelmä, mutta datapisteiden välisten etäisyyksien sijaan algoritmi perustuu datapisteiden esiintymistiheyden laskentaan. Toisin kuin K-means, DBSCAN ei ota klustereiden lukumäärää syötteenä. Sen sijaan klustereiden lukumäärää ohjaa naapuruston koko ja tiheys, jotka algoritmi ottaa syötteenään. Tähän mennessä käsiteltyjen osittavien menetelmien kohdalla on havaittu, että vaikkakin klusteroinnin avulla pystytään havaitsemaan mielenkiintoisia rakenteita aineistosta, eivät menetelmät ole kovin ansioituneesti kyenneet erottamaan poikkeavia havaintoja aineistosta. DBSCAN-menetelmällä on kuitenkin merkittävä etu tutkimusongelman kannalta verrattuna muihin käsiteltyihin ohjaamattomiin menetelmiin, sillä se ei aseta kaikkia aineiston datapisteitä klustereihin. Tämän ominaisuuden myötä muihin datapisteisiin nähden erillään sijaitsevat alkiot, joita ei sijoiteta mihinkään klusteriin, tulee luokiteltua poikkeavuuksiksi.

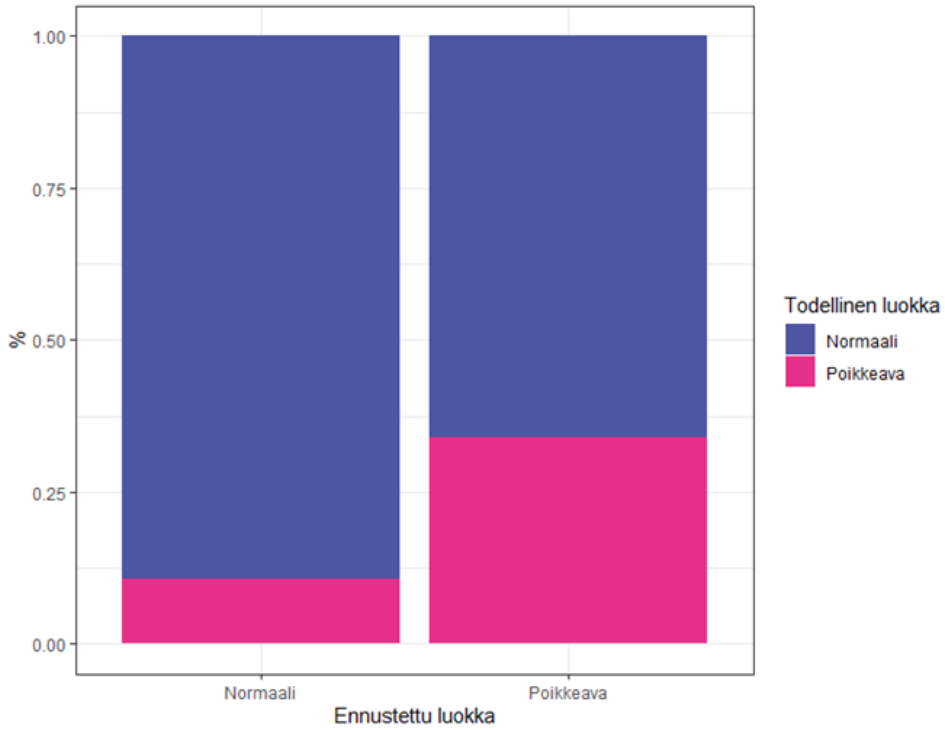
DBSCAN-algoritmin syöteparametrien arvot vaikuttavat klustereiden lukumäärään sekä kokoon. Mitä suurempi arvo naapuruston säteelle valitaan, sitä suurempi osa datapisteistä sijaitsee säteen sisällä ja näin ollen klustereihin asetettavien havaintojen lukumäärä kasvaa. Naapureiden lukumäärällä on samankaltainen vaikutus – mitä pienempi arvo valitaan, sitä enemmän klustereita muodostetaan ja tämän kautta poikkeavuuksiksi luokiteltavien datapisteiden lukumäärä vähenee. Koska tutkimuskysymyksenä on nimenomaan poikkeavien havaintojen tunnistus, halutaan tässä tapauksessa välttää naapuruston säteen asettamista liian suureksi ja toisaalta naapuruston lukumäärää liian pieneksi.

Naapureiden lukumäärän valitsemiseen ei ole kovin sofistikoituneita menetelmiä, vaan se usein määritetään aineiston rakenteen sekä olemassa olevan taustatiedon pohjalta. Naapuruston säteen sopivaa arvoa voidaan arvioida tarkastelemalla naapuruston sädettä vasten naapurustoon kuuluvien datapisteiden lukumäärää (kuva 28). 10 000 havainnon osajoukolla ja naapuruston eri lukumäärillä tarkasteltuna havaitaan polvikuvaajan taitekohdan asettuvan kohtaan, jossa naapuruston säteen arvo on kaksi. Naapuruston lukumäärä määritetään kokeilemalla aloittaen arvosta 100.



Kuva 28 Naapuruston säteeksi valikoituu ”polvi”-kuvaajan taitekohta

Sadalla naapurilla malli ennusti 44 % aineistosta poikkeavuuksiksi. Osuus on huomattavasti suurempi kuin aitojen poikkeavuuksien osuus aineistossa, joten syöteparametrien arvoja tulee muuttaa. Kun naapureiden vähimmäismääräksi asetetaan 50, ennustettu poikkeavuuksien osuus laskee 33 prosenttiin, joka on edelleen huomattavan suuri osuus. Naapureiden lukumäärä 12 tuottaa poikkeavuuksien osuudeksi 14.7 %, joka on hyvin lähellä poikkeavuuksien todellista osuutta ja näin ollen valikoituu lopullisen mallin parametrin arvoksi. Näillä valinnoilla DBSCAN muodostaa yhteensä 701 klusteria. Vertailtaessa ennustettuja poikkeavuuksia aineiston todellisiin poikkeaviin havaintoihin todetaan, että kolmannes ennustetuista poikkeavuuksista ovat todellisia poikkeavia havaintoja. Toisaalta muodostettujen klustereiden datapisteistä 11 % on todellisuudessa poikkeavuuksia (kuva 29), joten muodostettu DBSCAN-malli jättää poikkeavat havainnot 89 % todennäköisyydellä klustereiden ulkopuolelle.

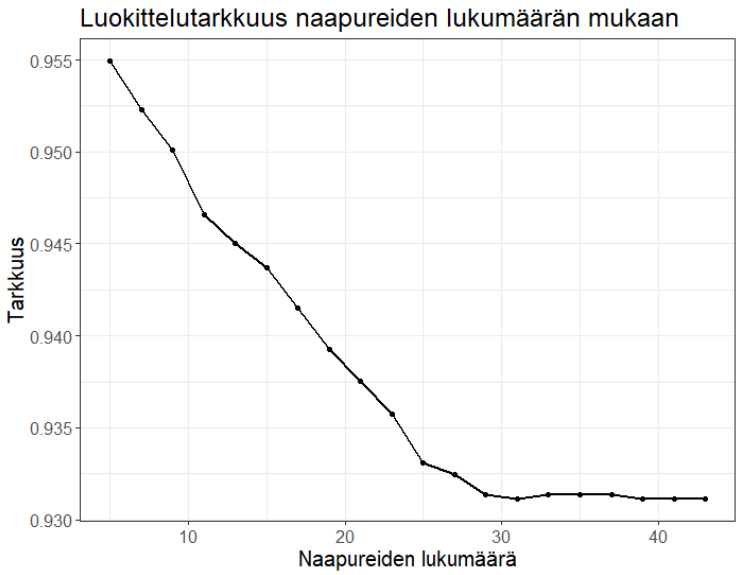


Kuva 29 Todellisten luokkien osuudet DBSCAN-mallin ennustamissa luokissa

5.5 k-NN

K:n lähimmän naapurin menetelmä on ohjatun oppimisen algoritmi, jossa opetusvaiheessa hyödynnetään tietoa datapisteiden oikeasta luokasta. Syöteparametrina algoritmille annetaan naapureiden lukumäärä, joka määrittää sen, kuinka monen lähimmän naapurin perusteella datapisteiden luokitus tehdään. Mallin yleistyskyvyn arvioimiseksi aineisto jaetaan opetus- ja testiaineistoon siten, että opetusaineistoon valitaan satunnaisesti 70 % aineiston havainnoista ja loput 30 % jätetään opetetun mallin suorituskyvyn testaamiseen. Osajoukkojen satunnainen poiminta on tärkeää, jotta molemmissa osajoukoissa ennustettavan luokan arvojen suhteelliset osuudet vastaavat koko aineiston osuuksia. Tässä tapauksessa sekä opetus- että testiaineistossa poikkeavien havaintojen osuus on 14 % kuten koko aineistossa.

Naapureiden optimaalinen lukumäärä etsitään opetusaineiston 10-osituksen ristiinvalidoinnilla tarkoittaen, että opetusaineisto jaetaan kymmeneen osajoukkoon ja vuorollaan kukin osa jätetään muun joukon ulkopuolelle testiaineistoksi. Opetusaineiston yhdeksää osaa käytetään mallin opettamiseen, ja ulkopuolelle jätetyllä testiaineistolla mitataan mallin tarkkuus. Näin saadaan etsittyä sellainen naapureiden lukumäärä, joka tuottaa korkeimman ennustetarkkuuden. Kuvasta 30 nähdään, että mitä suuremmaksi naapureiden lukumäärää kasvatetaan, sitä enemmän mallin tarkkuus kärsii. Ristiinvalidoinnin tuloksena lähimpien naapureiden lukumääräksi valikoituu k=5.



Kuva 30 Lähimpien naapureiden lukumäärä valitaan suurimman tarkkuuden mukaan

Mallin suorituskykyä arvioidaan luokittelemalla testiaineisto opetetulla mallilla ja vertaamalla ennustetun luokan arvoja testiaineiston todellisiin arvoihin. Viiden lähimmän naapurin mallin yleistyskyky on erinomainen sen onnistuessa luokittelemaan 99 % testiaineistosta oikein. Sekaannusmatriisista (taulukko 3) nähdään ennustettujen ja todellisten luokkien frekvenssit testiaineistossa ja niiden pohjalta voidaan laskea mallin tarkkuus, sensitiivisyys sekä spesifisyys. Mallin spesifisyys on hieman alhaisempi kuin sensitiivisyys, joka on sinänsä luonnollista poikkeavan luokan ollessa huomattavasti normaalia luokkaa pienempi. Käytännön tilastolaadinnan näkökulmasta alhaisempi spesifisyys sensitiivisyyteen nähden saattaa olla jopa tavoiteltavaa, koska tällöin suurempi osa poikkeavuuksiksi luokitelluista havainnoista on todellisuudessa poikkeuksellisia. Asiantuntijoiden täytyy viime kädessä arvioida poikkeuksellisiksi ennustetut havainnot ja tämä työ helpottuu, mikäli virheellisesti poikkeavuuksiksi ennustettujen normaaleiden havaintojen osuus on pieni.

Taulukko 3 Viiden lähimmän naapurin mallin tarkkuus on jopa 99 %

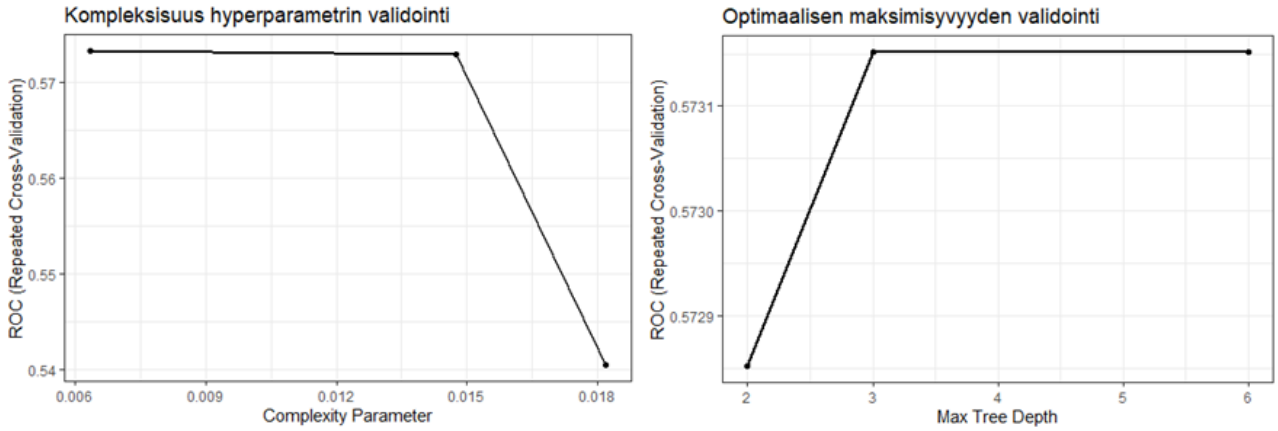
K:n lähimmän naapurin sekaannusmatriisi, k=5		
Todellinen luokka	**Ennustettu luokka**	
	Normaali	Poikkeava
Normaali	116 259	723
Poikkeava	576	18 324
Tarkkuus:	0.99	
Sensitiivisyys:	0.99	
Spesifisyys:	0.97	

5.6 Päätöspuu

Toisin kuin ohjaamattomat etäisyysmittoihin perustuvat klusterointimenetelmät sekä K:n lähimmän naapurin menetelmä, päätöspuu pystyy käsittelemään myös kategorisia muuttujia. Koska one hot –koodaus tuottaa harvoja (sparse) muuttujia, käytetään päätöspuumenetelmien kohdalla alkuperäisiä kategorisia muuttujia, jotta muuttujien merkitsevyys mallinnuksessa ei laskisi harvuuden seurauksena. Lisäksi aineistosta poistetaan voimakkaasti epätasapainoiset muuttujat, jotka kuvaavat raportoijakohtaisia tietoja kuten henkilöstön lukumäärää tai myönnettyjen luottokorttien lukumäärää. Tällaiset muuttujat saavat saman arvon jokaisella raportoijan raportoimalla rivillä, jonka vuoksi suuret raportoijat ohjaisivat luokittelua liiaksi. Näiden muuttujien poistolla varmistetaan, ettei poikkeavien havaintojen luokittelua tehdä raportoijalähtöisesti. Poikkeavuuksien tunnistukseen päätöspuumenetelmällä harkitaan kahta algoritmia, CART sekä C5.0, joille löytyy valmiit paketit R-ohjelmistossa.

5.6.1 CART

CART-menetelmällä puuta kasvatettaessa tulee ensimmäisenä löytää optimaaliset parametriarvot kompleksisuusparametrille sekä maksimisyvyydelle. Hyperparametrien arvot etsitään ristiinvalidoinnin avulla vastaavasti kuin lähimpien naapurien lukumäärä arvioitiin. Kuvasta 31 nähdään, että ristiinvalidoinnin lopputuloksena kompleksisuusparametrin optimaaliseksi arvoksi osoittautuu 0.015 ja, hieman yllättäen, optimaaliseksi maksimisyvyydeksi kolme.



Kuva 31 Kompleksisuuden ja puun maksimisyvyyden arviointi 10-osituksen ristiinvalidoinnilla

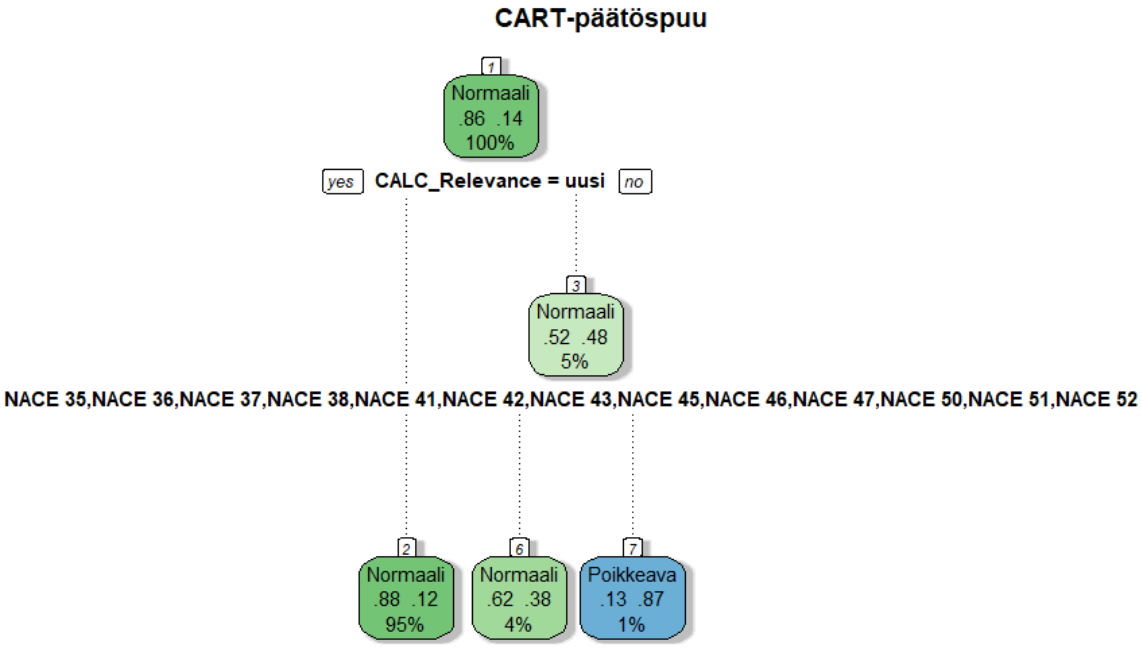
Lopullinen CART-puu opetetaan parametriarvoilla cp=0.015 ja maxdepth=3 ja aineistolla, joka tässä tapauksessa tarkoittaa 360 000 havaintoa vastaten 80 %:a periodin 2017M12 aineistosta. Puun suorituksen arviointiin käytetään testiaineistoksi jäljelle jäänyttä osajoukkoa, joka sisältää n. 90 000 havaintoa. Taulukon 4 sekaannusmatriisista nähdään, että kahden jaon päätöspuu pystyy hyvin luokittelemaan normaalit havainnot, mutta puu luokittelee myös selvän enemmistön poikkeavuuksista normaaliin luokkaan.

Taulukko 4 CART-puun sekaannusmatriisi, tarkkuus, sensitiivisyys sekä spesifisyys

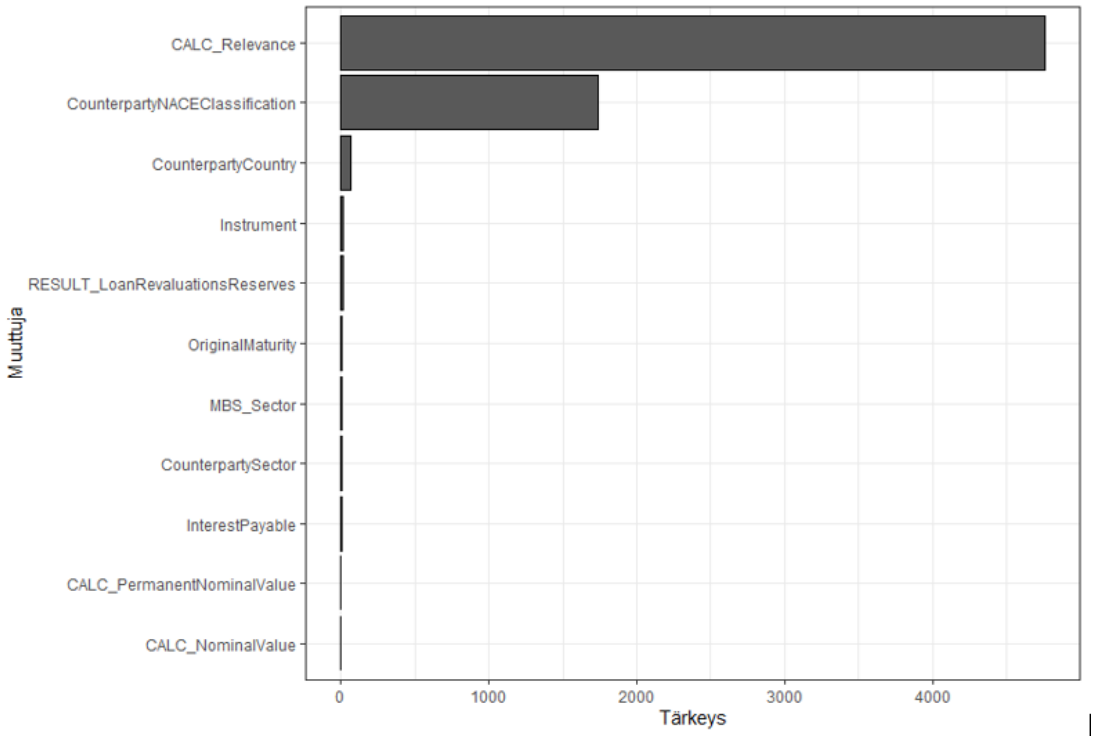
CART-puun sekaannusmatriisi		
Todellinen luokka	**Ennustettu luokka**	
	Normaali	Poikkeava
Normaali	77 774	145
Poikkeava	11 876	790
Tarkkuus:	0.87	
Sensitiivisyys:	0.87	
Spesifisyys:	0.84	

Päätöspuun rakennetta (kuva 32) ja muuttujien tärkeyttä (kuva 33) tarkasteltaessa huomataan, että luokittelun kannalta tärkein, puun juuresta tehtävä jako, tehdään muuttujan CALC_Relevance suhteen. Sen perusteella, onko havainto eli raportoitu rivi uusi, saadaan suoraan luokiteltua 95 % aineistosta normaaliin luokkaan. Tässä normaalissa luokassa 88 % datapisteistä on todellisia normaaleja havaintoja. Ne havainnot, jotka eivät ole raporteilla uusia, jaetaan seuraavaksi vastapuolen NACE- eli toimialaluokituksen mukaan. Mikäli vastapuoli kuuluu NACE-luokituksen luokkiin 85 - 98, luokitellaan se poikkeavaksi. Vaikkakin puu luokittelee

suurimman osan poikkeavuuksista normaaliin luokkaan, on poikkeavaksi luokiteltu havainto suurella todennäköisyydellä (87 %) todellisuudessa poikkeava. Muuttujien tärkeysjärjestystä tarkasteltaessa huomataan, että CALC_Relevance sekä NACE_luokka ovat selvästi muita muuttujia tärkeämpiä luokittelun näkökulmasta. Näiden jälkeen suurin luokittelukyky on vastapuolen maalla (CounterpartyCountry) sekä instrumentilla. Havainto on linjassa K-medoids-klusteroinnin tulosten kanssa, joiden osalta todettiin, että tärkein muuttuja normaalien ja poikkeavien havaintojen erottelussa oli CALC_Relevance.



Kuva 32 CART-puun rakenne

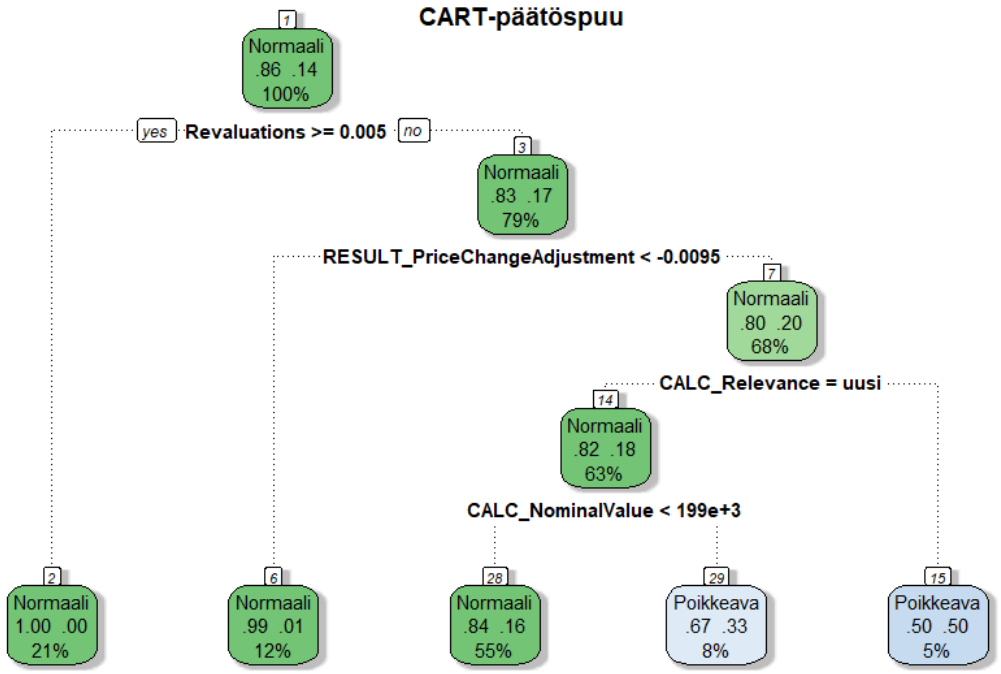


Kuva 33 CART-puun muuttujien tärkeysjärjestys

Poikkeavien havaintojen virheellistä luokittelua normaaliin luokkaan voidaan vähentää painottamalla spesifisyyttä yli sensitiivisyyden. Lisäämällä kolminkertainen paino väärin luokitelluille poikkeavuuksille (false negative) saadaan oikein luokiteltujen poikkeavuuksien osuutta kasvatettua kolmannekseen sensitiivisyyden noustessa 90 prosenttiin (taulukko 5), mutta toisaalta mallin tarkkuus kärsii väärin luokiteltujen normaalien havaintojen yleistyessä. Puun rakenne muuttuu painotuksen myötä, ja vasta kolmas jako tehdään painottamattoman puun tärkeimmän muuttujan suhteen (kuva 34). Käytännön kannalta ensimmäistä puuta, joka luokittelee hyvin pienen osuuden normaaleista havainnoista poikkeavuuksiksi, voidaan pitää parempana. Tunnistettaessa mahdollisia poikkeavuuksia tilastodatasta joutuu asiantuntija vahvistamaan mallin ehdottamat poikkeavuudet, jolloin suuri määrä ”vääriä hälytyksiä” lisää asiantuntijan työtaakkaa huomattavasti.

Taulukko 5 CART-puun sekaannusmatriisi, kun sensitiivisyydelle annetaan kolminkertainen paino

CART-puun sekaannusmatriisi		
Todellinen luokka	**Ennustettu luokka**	
	Normaali	Poikkeava
Normaali	70 891	7 028
Poikkeava	8 196	4 470
Tarkkuus:	0.83	
Sensitiivisyys:	0.89	
Spesifisyys:	0.39	

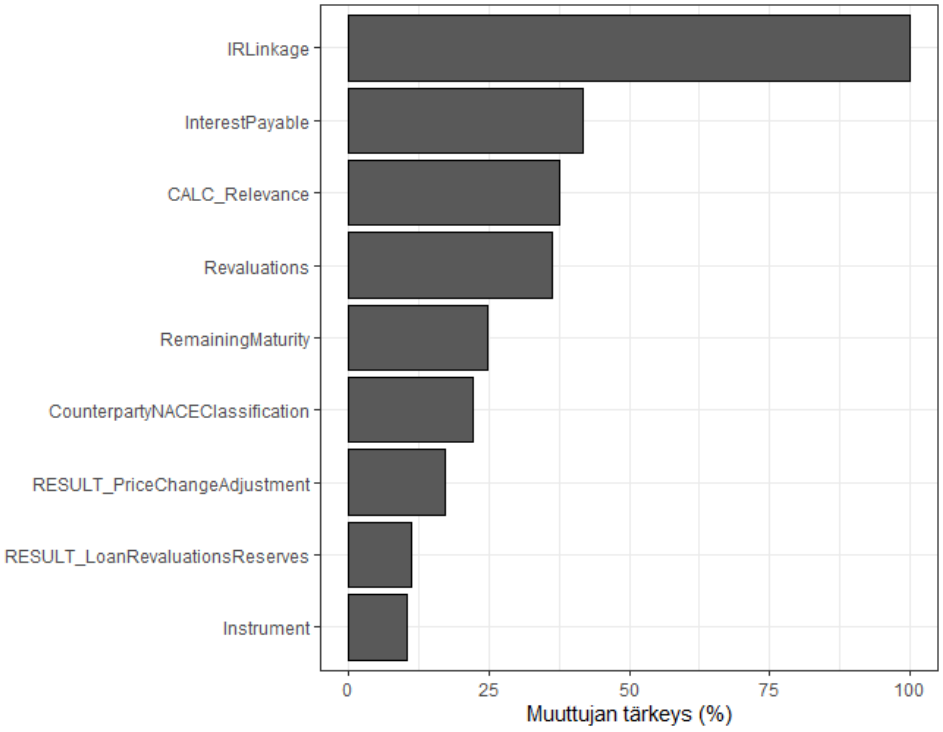


Kuva 34 CART-puun rakenne, kun sensitiivisyyden paino on kolminkertainen

5.6.2 C5.0

C5.0-algoritmin suurimmat erot CART-algoritmiin ovat jakokriteeri, joka C5.0:n tapauksessa on Gini-kertoimen tuottama Gini-lisä sekä solmujen jälkeläisten lukumäärä. C5.0 ei ole binääripuu kuten CART, vaan kahden jälkeläisen sijaan kullekin solmulle syntyy niin monta jälkeläistä, kuin jaon perusteena olevan muuttujan

arvoilla on luokkia. C5.0-puun kasvattamiseen käytetään CART-puun yhteydessä esikäsiteltyä aineistoa. Toisin kuin CART-puu, C5.0 ei ota syötteenään puun syvyyttä tai kompleksisuusparametria vaan puuta kasvatetaan, kunnes kaikki opetusaineiston havainnot on luokiteltu oikeisiin luokkiin. C5.0-puun muuttujien tärkeysprosentti kuvaa niiden havaintojen osuutta, jotka muuttujan suhteen jaetaan jossain puun solmussa. Näin ollen muuttuja, jonka suhteen juuresta tehdään jako, saa tärkeysprosentikseen 100. Muuttujien tärkeysjärjestystä tarkasteltaessa (kuva 35) nähdään, että luokittelun kannalta tärkeimmät muuttujat muodostavat jokseenkin eri muuttujajoukon kuin CART-puun tapauksessa. Puun tärkein muuttuja, jonka suhteen ensimmäinen jako tehdään, on korkosidonnaisuus (IRLinkage). Muita tärkeitä muuttujia ovat maksettava korko (InterestPayable) ja kuten CART-puussakin, myös CALC_Relevance on tärkeimpien muuttujien joukossa.



Kuva 35 C5.0-puun tärkeimmät muuttujat

Kun kasvatetun C5.0-puun yleistyskyky poikkeavuuksien luokittelussa testataan erillisellä testiaineistolla huomataan, että puu onnistuu luokittelemaan 91 % testiaineiston havainnoista oikein. Sekaannusmatriisista (taulukko 6) nähdään, että vaikka etenkin sensitiivisyys mutta myös spesifisyys saavat melko suuret arvot, ei puu vaikuta sopivalta mallilta tilastodatan poikkeavuuksien tunnistukseen. Syynä tähän on, että ennustettu poikkeavuusluokka sisältää lähes yhtä paljon virheellisesti luokiteltuja normaaleja havaintoja kuin todellisia poikkeavuuksia. Vaikka C5.0 on ennustetarkkuudeltaan CART-puuta parempi, on CART-puu tilastoaineiston poikkeavuuksien tunnistamiseen soveltuvampi malli.

Taulukko 6 C5.0-puu onnistuu ennustamaan oikean luokan 91 % todennäköisyydellä

C5.0-puun sekaannusmatriisi		
Todellinen luokka	**Ennustettu luokka**	
	Normaali	Poikkeava
Normaali	114 464	9 270
Poikkeava	2 488	9 656
Tarkkuus:	0.91	
Sensitiivisyys:	0.93	
Spesifisyys:	0.80	

5.7 Satunnaismetsä

Satunnaismetsä on päätöspuupohjainen menetelmä, joka CART-algoritmin tavoin pystyy käsittelemään sekä numeerisia että kategorisia muuttujia. Kuten päätöspuumallien kohdalla, myös satunnaismetsän osalta käytettiin aineistoa, jossa kategoriset muuttujat ovat mukana ilman one hot -käsittelyä. Ennen satunnaismetsän sovittamista muuttujiin täytyy kuitenkin tehdä joitakin muokkauksia, koska R-ohjelmiston RandomForest-funktio ei pysty käsittelemään sellaisia kategorisia muuttujia, jotka saavat yli 53 luokkaa. Aineistossa on neljä muuttujaa, joiden luokkien lukumäärä ylittää R-funktion ylärajan. Nimellisvaluutta (NominalCurrency), edellinen nimellisvaluutta (PREV_NominalCurrency) sekä vastapuolen kotimaa (CounterpartyCountry) käsitellään satunnaismetsää varten siten, että luokat, jotka sisältävät hyvin vähän havaintoja yhdistetään yhdeksi luokaksi. Näin saadaan luokkien lukumäärää redusoitua ilman, että kadotetaan olennaista informaatiota, kun yleisimmät luokat saadaan säilytettyä ennallaan. Vastapuolen NACE-luokitus käsitellään eri tavalla, koska havainnot jakautuvat tasaisemmin kaikkiin 100 eri luokasta. NACE-luokitukselle tehdään uusi muuttuja, johon yhdistetään alkuperäisen muuttujan luokkia siten, että yksi uusi luokka kattaa viisi alkuperäistä NACE-luokkaa. Esimerkiksi NACE-luokat 1-5 yhdistetään uudessa muuttujassa luokaksi 1-5. NACE-luokka 0 sisältää selvästi enemmän havaintoja kuin muut luokat, jonka vuoksi se jätetään omaksi luokakseen.

Satunnaismetsää sovittaessa tulee ensimmäisenä löytää sopivat hyperparametriarvot. Satunnaismetsän hyperparametreja ovat puiden lukumäärä sekä lukumäärä, montaako satunnaisesti valittua muuttujaa harkitaan kussakin jaossa. OOB-virhetermin perusteella optimaaliseksi muuttujien lukumääräksi osoittautuu kuusi. Ensimmäisen sovitettavan satunnaismetsän puiden lukumääräksi valitaan n=700 ja muuttujien lukumääräksi kuusi. Aineisto jaetaan opetus- ja testiaineistoksi siten, että molemmat osajoukot sisältävät 50 %

havainnoista. Näillä valinnoilla kasvatettu satunnaismetsä luokittelee opetusaineiston poikkeavuudet oikein lähes sata prosenttisesti ja normaaleistakin havainnoista vain 1 % luokitellaan virheellisesti poikkeavuuksiksi. Kun tällä mallilla ennustetaan testiaineiston luokka, ennustetarkkuus laskee huomattavasti (taulukko 7). Testiaineistolla poikkeavista havainnoista jopa yli puolet luokitellaan normaaleiksi, mutta normaalien havaintojen ennustetarkkuus säilyy korkeana.

Taulukko 7 Satunnaismetsän sekaannusmatriisit, kun mtry = 6 ja n = 700

Satunnaismetsän sekaannusmatriisi, opetusaineisto			**Satunnaismetsän sekaannusmatriisi, testiaineisto**		
Todellinen luokka	**Ennustettu luokka**		**Todellinen luokka**	**Ennustettu luokka**	
	Normaali	Poikkeava		Normaali	Poikkeava
Normaali	195 024	1 745	Normaali	191 704	3 331
Poikkeava	11	29 685	Poikkeava	17 523	13 907

Tarkkuus:	0.99	Tarkkuus:	0.91
Sensitiivisyys:	0.99	Sensitiivisyys:	0.98
Spesifisyys:	1	Spesifisyys:	0.44

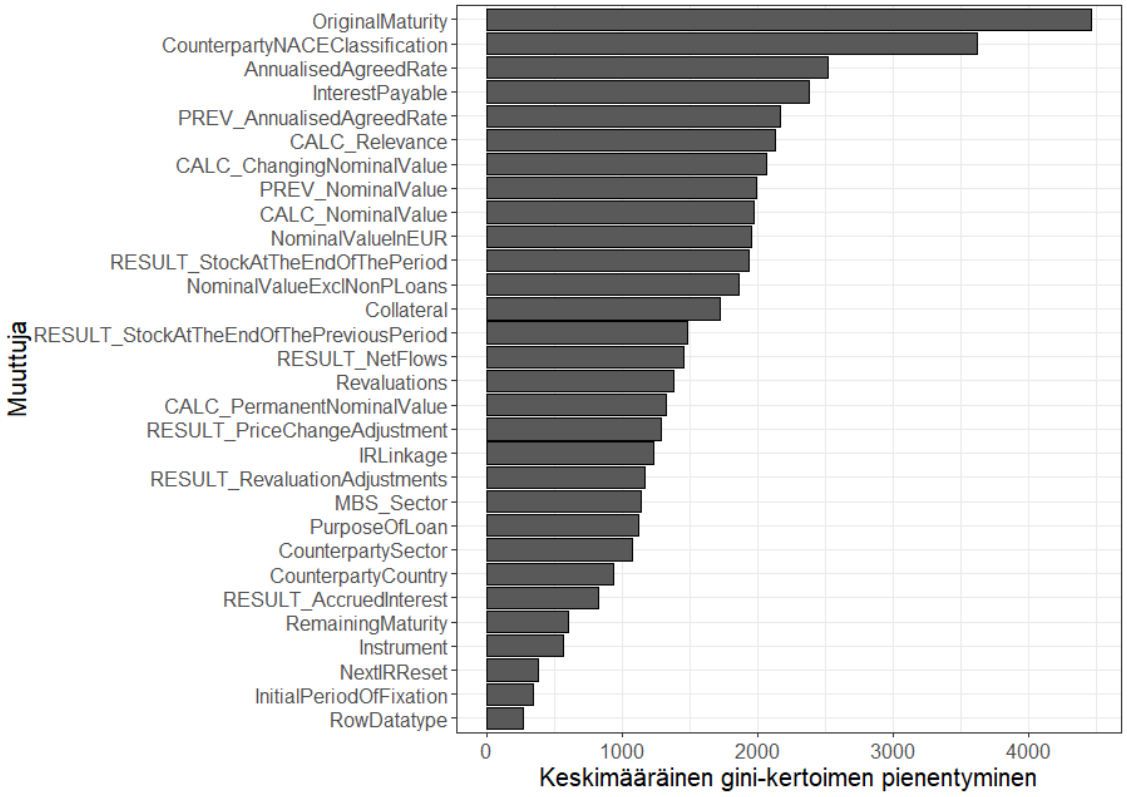
Ensimmäisen mallin tulos viittaa vahvasti ylisovitukseen, kun satunnaismetsän yleistyskyky testiaineistoon on huomattavasti heikompi kuin opetusaineistoon etenkin, kun katsotaan spesifisyysarvoa. Puiden lukumäärän kasvattamisen pitäisi ratkaista ylisovitusongelma, joten kasvatettavien puiden lukumäärä nostetaan tuhanteen. Lisäksi opetus- ja testiaineiston osuuksia muutetaan siten, että opetusaineistoon poimitaan 70 % koko aineiston havainnoista. Näiden muutosten jälkeen mallin ennustetarkkuus laskee hieman, mutta spesifisyys paranee huomattavasti 80 prosenttiin (taulukko 8). Tilastolaadinnan näkökulmasta ensimmäinen metsä on kuitenkin parempi, koska ennustettu poikkeavuusluokka sisältää huomattavasti vähemmän todellisia normaaleja havaintoja.

Taulukko 8 Satunnaismetsän sekaannusmatriisit, kun mtry = 6 ja n = 1000

Satunnaismetsän sekaannusmatriisi, opetusaineisto			**Satunnaismetsän sekaannusmatriisi, testiaineisto**		
Todellinen luokka	**Ennustettu luokka**		**Todellinen luokka**	**Ennustettu luokka**	
	Normaali	Poikkeava		Normaali	Poikkeava
Normaali	281 576	1 971	Normaali	115 078	11 013
Poikkeava	0	33 504	Poikkeava	1 944	7 843

Tarkkuus:	0.99	Tarkkuus:	0.90
Sensitiivisyys:	0.99	Sensitiivisyys:	0.91
Spesifisyys:	1	Spesifisyys:	0.80

Tarkasteltaessa muuttujien tärkeyttä luokittelun kannalta huomataan, että alkuperäinen maturiteetti on luokittelun kannalta tärkein muuttujan gini-kertoimella mitattuna. Gini-kertoimella mitattu muuttujan tärkeys kuvaa, kuin paljon muuttujan suhteen tehty jako tuottaa Gini-lisää (kuva 36). Kuten CART-algoritmilla kasvatetussa päätöspuussa, myös satunnaismetsän kohdalla vastapuolen toimialaluokitus (NACE-luokka) on tärkeimpien muuttujien joukossa. Sovittu vuosikorko on kolmanneksi tärkein muuttuja luokittelun kannalta. Myös edellinen sovittu vuosikorko (PREV_AnnualisedAgreedRate) on tärkeysjärjestyksessä korkealla. Havainto viittaa siihen, että ääriarvot sovitussa vuosikorossa on helppo tunnistaa laadunvalvontaprosessin visuaalisessa tarkastelussa ja näin ollen tulee korjattua herkästi. Toisaalta edellisen sovitun vuosikoron tärkeys luokittelun kannalta voi viitata siihen, että virheellisiä havaintoja ei heti korjata vaan ne korjaantuvat vasta seuraavilla periodeilla. Muuttujan tärkeys yhdessä viimeisimmän sovitun vuosikoron kanssa voi tarkoittaa myös sitä, että periodien väliset muutokset sovitussa vuosikorossa huomataan ja korjataan herkästi. CALC-Relevance, joka yksittäisten päätöspuiden osalta oli selvästi tärkein muuttuja, on satunnaismetsän muuttujien tärkeysjärjestyksessä vasta kuudentena.

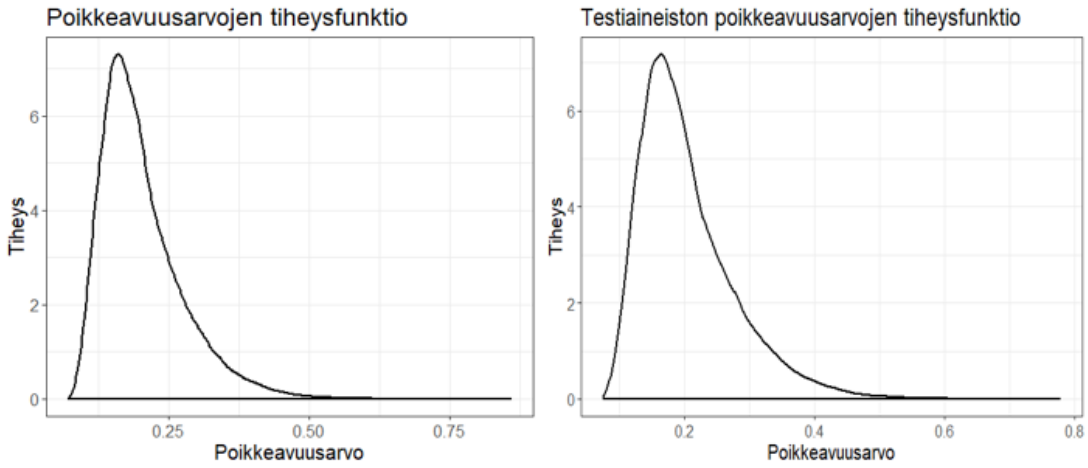


Kuva 36 Gini-kertoimen tuottama informaatiolisä muuttujittain

5.8 Eristysmetsä

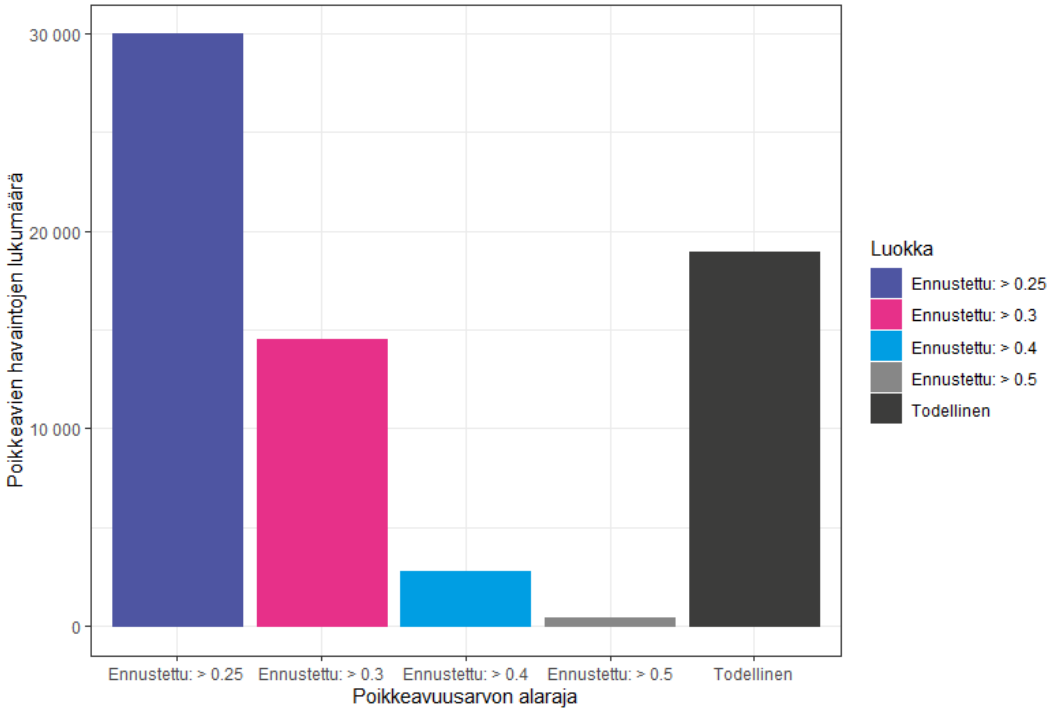
Eristysmetsä pystyy muiden päätöspuupohjaisten menetelmien tapaan käsittelemään sekatyyppistä aineistoa. Näin ollen eristysmetsän kasvattamiseen voidaan käyttää samaa aineistoa kuin satunnaismetsän kasvattamisessa käytettiin. Lisäksi eristysmetsää varten poikkeavuusluokka poistetaan aineistosta menetelmän perustuessa ohjaamattomaan oppimiseen, joka ei hyödynnä luokkatietoa opetusvaiheessa. Tietoa oikeasta luokasta voidaan kuitenkin käyttää hyväksi mallin tuloksia arvioidessa.

Eristysmetsä kasvatetaan opetusaineistolla, johon otetaan mukaan 70 % koko aineistosta. Jäljelle jäävälle testiaineistolle muodostetaan poikkeavuusarvot kasvatetun metsän mukaisesti. Poikkeavuusarvojen tiheysjakaumista nähdään, että opetusaineiston poikkeavuusarvot noudattavat lähes samaa jakaumaa kuin testiaineistolle lasketut poikkeavuusarvot (kuva 37). Poikkeavuusarvot saavat melko pieniä arvoja arvojen keskittyessä lähelle nollaa.



Kuva 37 Opetusaineiston poikkeavuusarvojen tiheysjakauma

Eristysmetsä tuottaa poikkeavuusarvon, mutta ei varsinaisesti luokittele datapisteitä poikkeavaan ja normaaliin luokkaan. Poikkeavuusarvon perusteella luokka voidaan muodostaa valitsemalla raja-arvo, jonka ylittävät poikkeavuusarvot luokitellaan poikkeaviksi ja sen alle jäävät normaaleiksi datapisteiksi. Kuten kuvasta 37 nähdään, poikkeavuusarvot ovat jakautuneet vinosti vasemmalle välillä [0,1]. Alkuperäisessä aineistossa poikkeavuuksia on noin 14 % ja jos tavoitteena on tunnistaa yhtä suuri osuus ennustettuja poikkeavuuksia, tulee arvon alarajaksi asettaa noin 0.3-0.5. Kuvasta 38 voidaan tulkita, että testiaineiston kohdalla poikkeavuusarvon asettaminen välille [0.25,0.3] ennustaisi suurin piirtein saman määrän poikkeavia havaintoja kuin niitä todellisuudessa on testiaineistossa.



Kuva 38 Poikkeavien havaintojen lukumäärät: Ennustetut vs. todelliset

Ainoastaan frekvenssien, tai suhteellisten osuuksien, perusteella ei kuitenkaan voida tehdä tarkkaa arviota siitä, mikä arvo poikkeavuusarvon alarajaksi olisi optimaalisin. Ennustetuista poikkeavuuksista sitä suurempi osa on todennäköisesti todellisuudessa normaaleja havaintoja, mitä pienemmäksi alaraja asetetaan. Tässä tapauksessa, kun myös oikeat luokat ovat tiedossa, voidaan raja-arvon valintaa arvioida myös sekaannusmatriisin avulla, kuten ohjatun oppimisen yhteydessä. Sekaannusmatriiseista (taulukko 9) nähdään, että korkeampi poikkeavuusarvon valinta johtaa korkeampaan tarkkuuteen, mutta samanaikaisesti spesifisyys kärsii. Toisaalta pienempi arvo tasoittaa matriisista laskettujen tunnuslukujen välisiä eroja, mutta tällöin ennustetarkkuus jää melko kehnoksi. Tuloksen voinee tulkita siten, että eristysmetsä suoriutuu hyvin yhden luokan ennustamisesta. Eristysmetsä yltää erittäin hyviin sensitiivisyysarvoihin, mutta jos pidetään tärkeänä sekä spesifisyyttä että sensitiivisyyttä, ei eristysmetsä voita vertailussa satunnaismetsää tai päätöspuuta. Tutkimusongelman valossa tarkasteltuna eristysmetsä voisi kuitenkin olla käyttökelpoinen menetelmä tilastodatan laadunvalvontaprosessissa. Eristyspuu mahdollistaa ennustettujen poikkeavuuksien määrän kontrolloinnin, joten poikkeavuusarvon alarajan valinnan kautta voidaan vaikuttaa siihen, kuinka paljon poikkeavia havaintoja tunnistetaan. Tämä voi olla hyödyllistä käytännön laadintaprosessin kannalta esimerkiksi erityisen kiireisinä aikoina, kun aikaa riittää vain suurimpien (tai todennäköisimpien) virheiden tarkistamiseen.

Taulukko 9 Eristysmetsän sekaannusmatriisit poikkeavuusarvon valinnan mukaan

Alkuperäisen aineiston luokka	**Ennustettu luokka: s(x,N) > 0.4**	
	Normaali	Poikkeava
Normaali	115 230	1 738
Poikkeava	17 854	1 057
	Tarkkuus:	0.86
	Sensitiivisyys:	0.98
	Spesifisyys:	0.06
Alkuperäisen aineiston luokka	**Ennustettu luokka: s(x,N) > 0.3**	
	Normaali	Poikkeava
Normaali	106 520	10 448
Poikkeava	14 867	4 044
	Tarkkuus:	0.81
	Sensitiivisyys:	0.91
	Spesifisyys:	0.21
Alkuperäisen aineiston luokka	**Ennustettu luokka: s(x,N) > 0.2**	
	Normaali	Poikkeava
Normaali	71 655	45 313
Poikkeava	6 294	12 617
	Tarkkuus:	0.62
	Sensitiivisyys:	0.61
	Spesifisyys:	0.67

6. YHTEENVETO

Tutkielmassa harkittiin kahdeksaa eri koneoppimismenetelmää kaksiluokkaiseen luokittelutehtävään, poikkeavien havaintojen tunnistamiseen. Ohjaamattoman oppimisen menetelmistä K-means-klusterointi ei onnistunut muodostamaan klustereita, jotka osittaisivat aineistoa poikkeavuuden perusteella vaan sen sijaan poikkeavat havainnot jakautuivat tasaisesti kaikkiin klustereihin. Klusteroinnin avulla kuitenkin löydettiin aineistosta kiinnostavia rakenteita, joita voidaan hyödyntää datan analysoinnissa. Lisäksi voisi olla hyödyllistä huomioida löydettyjä klustereita laadunvalvontaprosessissa esimerkiksi ryhmittelemällä saapuvaa aineistoa klustereiden pohjalta ja kohdentamalla laadunvalvontaa eri osajoukoille. K-means++ ja K-medoids tuottivat samankaltaisia tuloksia, mutta onnistuivat myös jokseenkin paremmin poikkeavien havaintojen erottelussa muusta datajoukosta. Molemmilla menetelmillä syntyi yksi klusteri, joka sisälsi suhteellisesti enemmän poikkeavuuksia kuin muut klusterit. Lisäksi molemmilla menetelmillä syntyi klusteri, joka ei sisältänyt juuri lainkaan poikkeavuuksia. Tiheysperustainen klusterointi tuotti tulkinnaltaan melko samankaltaisia tuloksia kuin K-means++ ja K-medoid. DBSCAN-klusteroinnin tuloksena normaalit havainnot onnistuttiin luokittelemaan hyvin, mutta ennustettu poikkeavuusluokka sisälsi melko paljon myös todellisia normaaleja datapisteitä. Tuloksista voidaan vetää johtopäätös, että ohjaamattoman oppimisen osittavat menetelmät onnistuvat hyvin erottamaan poikkeavat havainnot normaaleista havainnoista, mutta toisaalta samanaikaisesti todellisia normaaleja havaintoja tulee luokiteltua poikkeavuuksiksi.

Yksittäiset päätöspuut, CART ja C5.0, suoriutuivat luokittelutehtävästä melko hyvin. Vaikkakin CART-algoritmi luokitteli suuren osan poikkeavista havainnoista normaaliin luokkaan, koostui ennustettu poikkeava luokka todellisista poikkeavista havainnoista. C5.0-algoritmi sitä vastoin onnistui erityisesti normaalin luokan ennustamisessa, mutta ennustetusta poikkeavasta luokasta jopa puolet koostui todellisista normaaleista havainnoista. C5.0-algoritmin voidaankin todeta suoriutuvan hyvin normaalien havaintojen tunnistuksessa, mutta ei niinkään poikkeavien havaintojen. CART-algoritmin suoriutuminen poikkeavien havaintojen tunnistuksessa on käytännön laadunvalvonnan näkökulmasta melko mielekäs, koska poikkeaviksi havainnoiksi luokitellut havainnot ovat pääasiassa todellisia poikkeavuuksia, eikä tällöin laadintaprosessia kuormita ”väärien hälytysten” tarkistaminen.

Satunnaismetsä suoriutui luokittelusta CART-puun tavoin, mutta mallin ennustetarkkuus oli, odotusten mukaisesti, CART-puuta parempi. Satunnaismetsä onnistui tunnistamaan normaalit havainnot 98 % tarkkuudella, mutta toisaalta kohtuullinen osuus myös todellisista poikkeavista havainnoista ennustettiin normaaleiksi. Kuten edellä todettiin, tämä on kuitenkin erinomainen tulos tilastojen laadunvalvonnan näkökulmasta ennustetun poikkeavan luokan ollessa verrattain ”puhdas”.

Päätöspuupohjaisista algoritmeista viimeisenä tarkasteltiin erityisesti poikkeavien havaintojen tunnistukseen kehitettyä ohjaamattoman oppimisen algoritmia, eristysmetsää. Kuten CART ja satunnaismetsä, onnistui eristysmetsä ennustamaan tarkasti normaalit havainnot, mutta jos kaikki poikkeavat havainnot haluttaisiin tunnistaa, sisältäisi ennustettu poikkeavuusluokka merkittävän paljon myös todellisia normaaleja havaintoja. Kuitenkin todellisten poikkeavuuksien ennustamiseen eristysmetsä on soveltuva menetelmä, jos ei ole kriittistä tunnistaa kaikkia todellisia poikkeavuuksia. Poikkeavuusarvon alarajaa korottamalla saadaan luokiteltua kaikkein todennäköisimmät poikkeavuudet, ja normaalien havaintojen osuus poikkeavuusluokassa pienenee nopeasti alarajaa korottamalla.
Harkituista menetelmistä poikkeavien havaintojen tunnistuksesta ylivoimaisesti parhaiten suoriutui ohjattuun oppimiseen perustuva k:n lähimmän naapurin menetelmä, joka onnistui ennustamaan sekä normaalit että poikkeavat havainnot lähes täydellisesti. K:n lähimmän naapurin menetelmällä ennustettuna poikkeavuusluokka paitsi sisältää pääasiassa vain todellisia poikkeavia havaintoja, sisältää myös lähes kaikki todelliset poikkeavuudet.
Koneoppimismenetelmien avulla onnistuttiin tunnistamaan poikkeavat havainnot tilastollisesta aineistosta hyvin sekä löydettiin mielenkiintoisia rakenteita aineistosta. Lisäksi malleja arvioidessa havaittiin, että erityisesti havainnon aggregointitasoa kuvaava muuttuja sekä vastapuolen toimialaluokitusta kuvaava muuttuja ovat tärkeitä erotellessa poikkeavia havaintoja muusta aineistosta. Näiden lisäksi sovittu vuosikorko, maturiteetti, vastapuolen maa sekä taloustoimi osoittautuivat tärkeiksi muuttujiksi luokittelun kannalta. Jokseenkin odottamattomasti jatkuvat muuttujat, kuten markkina-arvo, eivät nousseet luokitteluissa esiin.

Suurin osa menetelmistä ennusti normaalin luokan suurella tarkkuudella tarkoittaen, että ennustetut poikkeavat luokat sisälsivät vain vähän havaintoja, jotka eivät ole todellisia poikkeavuuksia. Toisaalta tällöin myös merkittävä osa todellisista poikkeavista havainnoista ennustettiin normaaleiksi havainnoiksi. Tämä ominaisuus on kuitenkin tilastolaadinnan kannalta oikein hyvä, sillä tilastoja tarkastellaan ja analysoidaan aggregoidulla tasolla, jolloin vain aggregaattiin vaikuttavat poikkeavuudet on mielekästä korjata. Tilastolaadinnassa ei pääsääntöisesti ole tarkoituksen mukaista korjata sellaisia havaintoja, joilla ei ole suurta merkitystä aggregaattitason tilastoon. Tällaisten havaintojen korjaaminen lähinnä kuormittaisi sekä raportoijia että tilastolaatijoita ilman merkittävää hyötyä tilaston ja sen analysoinnin kannalta.

Tämän tutkielman tulosten varjolla voidaan todeta, että koneoppimismallien hyödyntäminen osana tilastolaadintaa tuottaisi prosessiin lisäarvoa laadunvalvonnan näkökulmasta. Lisäksi tulokset viittaavat siihen, että klusterointimenetelmien käyttö analyysityön tukena auttaisi hahmottamaan tilastollisesta aineistosta rakenteita, joita on vaikeaa tunnistaa aineiston kaksiulotteisissa tarkasteluissa. Tulevaisuudessa olisi kiintoisaa myös tarkastella tässä tutkielmassa muodostettujen mallien tuloksia raportoijatasolla, jolloin nähtäisiin, kuvaavatko syntyneet klusterit joitakin yksittäisiä raportoijia tai raportoijaryhmiä. Tällaisesta analyysista voisi

olla suuri hyöty laadunvalvontaprosessin tehokkuuden kannalta, kun laadunvalvontavastuita voitaisiin ohjata tulosten mukaan siten, että aineistoltaan samankaltaisia raportoijia tarkasteltaisiin osajoukkoina erillään muusta aineistosta. Lisäksi, koska koneoppimismallit ovat hyvin dataspesifejä, olisi mielenkiintoista kartoittaa tutkielmassa käytettyjen menetelmien käyttökelpoisuutta myös muilla rahoitustilastoaineistoilla. Tällöin nähtäisiin, soveltuvatko tässä tutkielmassa hyviksi todetut menetelmät myös muiden tilastollisten aineistojen poikkeavuuksien tunnistamiseen.

LÄHTEET

Arthur, D. V. (2007). K-Means++: The Advantages of Careful Seeding. *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms.* doi:10.1145/1283383.1283494

Bernard, S. H. (2010). A Study of Strength and Correlation in Random Forests. *Conference: Advanced Intelligent Computing Theories and Applications*, 186-191. doi:10.1007/978-3-642-14831-6_25

Breiman, L. (1996). OUT-OF-BAG Estimation. Noudettu osoitteesta

https://pdfs.semanticscholar.org/e408/af07b74476564e3d5511ed6b169fa1f2a484.pdf

Breiman, L. (2001). Random Forests. *Machine Learning*, 5-32. doi:https://doi.org/10.1023/A:1010933404324

Breiman, L. F. (1984). *Classification and Regression Trees.* Noudettu osoitteesta https://books.google.fi/books?id=MGlQDwAAQBAJ&printsec=frontcover&hl=fi#v=onepage&q&f=false

Chandola, V. K. (2009). Outlier Detection: A Survey. *ACM Computing Surveys*. doi:10.1145/1541880.1541882

Chapelle, O. S. (2006). *Semi-Supervised Learning.* The MIT Press. Noudettu osoitteesta http://www.acad.bg/ebook/ml/MITPress-%20SemiSupervised%20Learning.pdf

Chomboon, K. C. (2015). An Empirical Study of Distance Metrics for k-Nearest Neighbor Algorithm. *International Conference on Industrial Application Engineering 2015.* doi:10.12792/iciae2015.051

Cunningham, P. D. (2007). *k-Nearest Neighbour Classifiers.* Noudettu osoitteesta https://www.researchgate.net/publication/228686398_k-Nearest_neighbour_classifiers

Dietterich, T. (1999). An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. *Machine Learning*, 1–22. Noudettu osoitteesta http://web.engr.oregonstate.edu/~tgd/publications/mlj-randomized-c4.pdf

Drummond, C. H. (2000). *Exploiting the Cost (In)sensitivity of Decision Tree Splitting Criteria.* From: AAAI Technical Report. Noudettu osoitteesta https://www.aaai.org/Papers/Workshops/2000/WS-00-05/WS00-05-009.pdf

Dudani, A. (1976). The Distance-Weighted k-Nearest-Neighbor Rule. *IEEE Transactions on Systems, Man, and Cybernetics*, 325-327. doi:10.1109/TSMC.1976.5408784

Ester, M. K.-P. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Noudettu osoitteesta https://pdfs.semanticscholar.org/179d/92446a495d1d00a5d7d37de73a578f8db459.pdf

Freund, Y. S. (1996). Experiments with a New Boosting Algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference*. Noudettu osoitteesta https://cseweb.ucsd.edu/~yfreund/papers/boostingexperiments.pdf

Hartigan, J. W. (1979). Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 100-108. doi:10.2307/2346830

Hawkins, D. (1980). *Identification of Outliers.*

Hssina, B. M. (2010). A comparative study of decision tree ID3 and C4.5. *International Journal of Advanced Computer Science and Applications*. doi:10.14569/SpecialIssue.2014.040203

Huang, Z. L. (2012). Anomaly detection in clinical processes. *AMIA Annu Symp Proc*, 370–379. doi:10.4338/ACI-2015-05-RA-0054

Ijaz, M. A. (2018). Hybrid Prediction Model for Type 2 Diabetes and Hypertension Using DBSCAN-Based Outlier Detection, Synthetic Minority Over Sampling Technique (SMOTE), and Random Forest. doi:https://doi.org/10.3390/app8081325

Kaufman, L. R. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis.* John Wiley & Sons.

Klawonn, F. (2016). Exploring Data Sets for Clusters and Validating Single Clusters. *Procedia Computer Science*, 1381-1390. doi:https://doi.org/10.1016/j.procs.2016.08.183

Kotsiantis, S. (2007). Supervised Machine Learning: A Review of Classification Techniques. *Emerging Artificial Intelligence Applications in Computer Engineering* (ss. 3-24). IOS Press.

Kurien, L. (2019). Detection and prediction of credit card fraud transactions using machine learning. *International journal of engineering sciences & research technology*. doi:10.5281/zenodo.2608242

Liu, B. X. (2005). Clustering Via Decision Tree Construction. *Studies in Fuzziness and Soft Computing*. Noudettu osoitteesta http://web.cs.ucla.edu/~wwc/course/cs245a/CLTrees.pdf

Liu, F. T. (2012). Isolation-based Anomaly Detection. doi:10.1145/2133360.2133363

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 129-137. Noudettu osoitteesta https://sites.cs.ucsb.edu/~veronika/MAE/kmeans_LLoyd_Least_Squares_Quantization_in_PCM.pdf

MacQueen, J. (1967). *Some Methods for Classification and Analysis of Multivariate Observations.* Noudettu osoitteesta https://www.semanticscholar.org/paper/Some-methods-for-classification-and-analysis-of-MacQueen/ac8ab51a86f1a9ae74dd0e4576d1a019f5e654ed

Mingers, J. (1989). An Empirical Comparison of Pruning Methods for Decision Tree Induction. *Machine Learning 4*, 227-243. doi:10.1023/A:1022604100933

Patel, N. U. (2012). Study of Various Decision Tree Pruning Methods with. *International Journal of Computer Applications*. doi:10.5120/9744-4304

Patil, S. N. (2018). Predictive Modelling For Credit Card Fraud Detection Using Data Analytics. *Procedia Computer Science, 132*, 385-395. doi:https://doi.org/10.1016/j.procs.2018.05.199

Phua C, Lee VCS, Smith-Miles K, Gayler RW. (2010). A comprehensive survey of data mining-based fraud detection research. *arXiv:1009.6119*.

Quinlan, J. (1986). Induction of Decision Trees. *Machine Learning*. doi:https://doi.org/10.1023/A:1022643204877

Quinlan, J. (2014). *C4.5: Programs for Machine Learning.* Elsevier.

Song, Y. L. (2015). Decision tree methods: applications for classification and prediction. *Shanghai Arch Psychiatry*, 130-135. doi:10.11919/j.issn.1002-0829.215044

Stanfill, C. W. (1986). Toward Memory-Based Reasoning. *Communications of the ACM*. doi:10.1145/7902.7906

Sutton, R. B. (2017). Reinforcement Learning: An Introduction. Teoksessa *A Bradford Book.* The MIT Press. Noudettu osoitteesta http://incompleteideas.net/book/bookdraft2017nov5.pdf

Taboada-Crispi, A. S.-P. (2009). Anomaly Detection in Medical Image Analysis. doi:10.4018/978-1-60566-314-2.ch027

Wilson, D. M. (1997). Improved Heterogeneous Distance Functions. *Journal of Artificial Intelligence Research*. Noudettu osoitteesta https://www.jair.org/index.php/jair/article/view/10182/24168

Zhang, Y. M. (2010). Outlier Detection Techniques for Wireless Sensor Networks: A Survey. *IEEE Communications Surveys & Tutorials*. doi:10.1109/SURV.2010.021510.00088